OTRS
Open Technology
Real Services

Documentation

OTRS 3.3 - 管理者マニュアル

Build Date:

2013-04-30

OTRS 3.3 - 管理者マニュアル

製作著作 © 2003-2013 OTRS AG

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

この著作はOTRS AGによって著作権で保護されています。

コピーがこの著作権宣言文を保持する限り、全体あるいは一部分をコピーしても良いです。

このドキュメントのソース・コードはgithubのリポジトリdoc-adminで見つけることができます。

UNIX は X/Open Company Limited. の登録商標です。 Linux は Linus Torvalds の登録商標です。

MS-DOS, Windows, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows 2003, WindowsVista, Windows 7 は、Microsoft Corporation の登録商標です。その他の商標及び登録商標: SUSE と YaST は SUSE Linux GmbH の, Red Hat と Fedora は Red Hat, Inc の登録商標です。 Mandrake は MandrakeSoft, SA の登録商標です。Debian は Software in the Public Interest, Inc の登録商標です。MySQL と the MySQL Logo は Oracle Corporation と、あるいは、その系列企業の登録商標です。

全ての登録名は、無償利用の保障なしに使用されており、そして恐らくそれらは登録商標です。

OTRS AGは、基本的にメーカーの記法に従います。その他の製品は各製造者の登録商標で記載しています。

目次

序文 viii
1. 導入 1
    1. トラブル・チケット・システム - 基礎 1
        1.1. トラブル・チケット・システムとは何でしょうか？なぜあなたはそれを必要としているのでしょうか？ 1
        1.2. トラブル チケットとは？ 2
    2. OTRS ヘルプデスク 2
        2.1. 基礎 2
        2.2. 機能 2
        2.3. ハードウェアとソフトウェアの必要条件 11
        2.4. コミュニティー 12
        2.5. OTRSのためのプロフェッショナル・サービス 12
2. インストレーション 13
    1. シンプルな方法 - あらかじめ構築されたパッケージのインストレーション 13
        1.1. SUSE Linuxサーバ上でのRPMのインストール 13
        1.2. Red Hat Enterprise Linux あるいは CentOS システムへのOTRSのインストール 15
        1.3. Debian または Ubuntuシステム上でのOTRSのインストール 20
        1.4. Microsoft Windows システム上でのOTRSのインストール 20
    2. ソース (Linux,Unix)からのインストレーション 22
        2.1. ソースからのインストレーションの準備 22
        2.2. Perlモジュールのインストレーション 23
        2.3. Apacheウェブサーバの構成 24
        2.4. データベースの構成 25
        2.5. OTRS用cronジョブのセットアップ 26
        2.6. OTRSスケジューラ・サービス 29
    3. ウェブ・インストーラの使用 29
    4. OTRSフレームワークのアップグレード 33
    5. Windowsインストーラのアップグレード 37
        5.1. Windows上でパッチ・レベルのアップグレードを実施します。 37
        5.2. Windows上でのマイナー・レベルのアップグレードの実施 38
    6. 追加のアプリケーション 38
        6.1. FAQ (よくある質問) 38
3. ファースト・ステップ 39
    1. 担当者ウェブインターフェイス 39
    2. 顧客ウェブインターフェイス 39
    3. 公開ウェブインターフェイス 40
    4. はじめてのログイン 40
    5. ウェブインターフェイス - 概要 41
    6. ダッシュボード 43
    7. キューとは何か？ 46
    8. キュー概要とは何か？ 47
    9. ユーザ選択 (選好) 48
4. 管理 50
    1. OTRSの管理者エリア 50
        1.1. 基礎 50
        1.2. 担当者、グループおよび役割 50
        1.3. 顧客および顧客グループ 59
        1.4. キュー 62
        1.5. 挨拶文、署名、添付およびテンプレート 64
        1.6. 自動応答 72
        1.7. Eメール・アドレス 75
        1.8. 通知 76
        1.9. SMIME 77
        1.10. PGP鍵 78

1.11. 状態 78
1.12. SysConfig 79
1.13. Eメールアカウントの使用 80
1.14. 受信Eメール・メッセージのフィルタリング 81
1.15. GenericAgentでの自動仕事の実行 84
1.16. 管理者Eメール 85
1.17. セッション管理 86
1.18. システムログ 87
1.19. SQLボックス経由でのSQLクエリー 87
1.20. パッケージ・マネージャー 88
1.21. ウェブサービス 89
1.22. 動的フィールド 89
2. システム構成 90
2.1. OTRS構成ファイル 90
2.2. ウェブインターフェースによるシステムの構成 91
3. システムのバックアップ 92
3.1. バックアップ 92
3.2. リストア 92
4. Eメール設定 93
4.1. Eメールの送信/受信 93
4.2. PGP鍵を備えた安全なEメール 98
4.3. S/MIMEを備えた安全なEメール 100
5. 外部バックエンドの使用 103
5.1. 顧客情報 103
5.2. 顧客ユーザ・バックエンド 104
5.3. 担当者と顧客を認証するバックエンド 111
5.4. 顧客自己登録のカスタマイズ 115
6. チケット設定 117
6.1. チケットの状態 117
6.2. チケット・プロパティ 120
6.3. チケット責任者とチケットの監視 120
7. 時間関連機能 123
7.1. 業務時間、祝祭日とタイムゾーンの設定 123
7.2. 自動アンロック 124
8. PDF出力のカスタマイズ 124
9. 統計モジュール (Stats module) 125
9.1. 担当者によるモジュールの取り扱い 125
9.2. OTRS管理者による統計モジュールの管理 134
9.3. システム管理者による統計モジュールの管理 134
10. 動的フィールド 136
10.1. 導入 136
10.2. 構成 137
11. 一般インターフェイス 155
11.1. 一般インターフェース・レイヤー 155
11.2. ウェブサービス・コマンドライン・インタフェース 157
11.3. ウェブサービス 160
11.4. ウェブサービス・グラフィカル・インターフェース 160
11.5. ウェブサービス・コマンドライン・インターフェイス 176
11.6. ウェブサービス構成 177
11.7. コネクタ 184
12. OTRSスケジューラ 195
12.1. スケジューラ・グラフィカル・インターフェース 195
12.2. スケジューラ・コマンドライン・インタフェース 197
5. カスタマイズ 201
1. アクセス・コントロール・リスト(ACL) 201
1.1. 導入 201
1.2. サンプル 201
1.3. 参照 203

2. 工程管理 206
2.1. 導入 206
2.2. サンプル・プロセス 206
2.3. 例の実行 207
2.4. プロセス構成リファレンス 231
3. 自分のテーマの作成 249
4. OTRSフロント・エンドのローカライゼーション 250
6. パフォーマンス・チューニング 251
1. OTRS 251
1.1. TicketIndexModule 251
1.2. TicketStorageModule (チケット・格納・モジュール) 251
1.3. チケットのアーカイブに保管 252
2. データベース 253
2.1. MySQL 253
2.2. PostgreSQL 253
3. ウェブサーバ 253
3.1. 事前に確立しているデータベース接続 253
3.2. プレインストールされたモジュール - startup.pl 253
3.3. ディスク上で更新された時パール・モジュールをリロードします。 253
3.4. 正しい戦略の選択 254
3.5. mod_gzip/mod_deflate 254
A. 追加の資料 255
1. OTRSグループのウェブサイト 255
2. メーリングリスト 255
3. バグ・トラッキング 255
4. 商用サポート 256
B. Configuration Options Reference 257
1. DynamicFields 257
2. Framework 262
3. GenericInterface 392
4. ProcessManagement 404
5. Scheduler 412
6. Ticket 415
C. 敬意と賞賛 646
D. GNUフリー文書許諾契約書 648
0. はじめに 648
1. この利用許諾契約書の適用範囲と用語の定義 648
2. 逐語的に忠実な複製 649
3. 大量の複製 649
4. 改変 650
5. 文書の結合 651
6. 文書の収集 651
7. 独立した著作物の集積 652
8. 翻訳 652
9. 契約の終了 652
10. 将来における本利用許諾契約書の改訂 652
この利用許諾契約書をあなたの文書に適用するには 652

表の一覧

2.1. いくつかのcronジョブ・スクリプトの記述 27
4.1. デフォルト・グループは新規のOTRSインストールで利用可能です 51
4.2. OTRSグループと関連した権限 54
4.3. 追加の許可グループ 54
4.4. 自動応答用イベント 73
4.5. 異なるX-OTRSヘッダーの機能 81
4.6. 次のフィールドがシステムへ追加されます。: 139
4.7. テーブル4.7. init.d スクリプトのリストとサポートされるオペレーティング・システム . 197
A.1. メーリングリスト 255

例の一覧

4.1. 特定のキューの中へのスパム・メールのソート 83
4.2. .fetchmailrc 95
4.3. フィルタ・モジュールのためのサンプルジョブ 96
4.4. フィルタ・モジュールのためのサンプル・ジョブ
Kernel::System::PostMaster::Filter::CMD 96
4.5. フィルター・モジュールのためのサンプル・ジョブ
Kernel::System::PostMaster::Filter::ExternalTicketNumberRecognition 97
4.6. DB顧客バックエンドの構成 104
4.7. DBバックエンドを備えた会社チケットの使用 106
4.8. LDAP顧客バックエンドの構成 107
4.9. LDAPバックエンドを備えた会社チケットの使用 108
4.10. OTRSでひとつ以上の顧客バックエンドを使用する 109
4.11. DBバックエンドに対する認証エージェント 111
4.12. LDAPバックエンドに対する認証エージェント 111
4.13. HTTPBasicを使用して、担当者を認証します。 113
4.14. Radiusバックエンドに対する担当者を認証します。 113
4.15. DBバックエンドに対する顧客ユーザ認証 113
4.16. LDAPバックエンドに対する顧客ユーザ認証 113
4.17. HTTPBasicを備えた顧客ユーザ認証 114
4.18. Radiasのバックエンドに対する顧客ユーザ認証 114
4.19. 値系列の定義 - 1つの要素 132
4.20. 値系列の定義 - 2つの要素 132
4.21. 新規電話チケット画面にあるフィールド 1 を活性化します。 148
4.22. 新規電話チケット画面中の必須フィールド1を活性化 149
4.23. 必須として新規電話チケット画面中のフィールド1を活性化 149
4.24. 新規電話チケット画面中の複数の必須フィールドを非活性化 150
4.25. チケットズーム画面でフィールド 1 を活性化 151
4.26. チケット概要小画面の中のField1を活性化 152
4.27. チケット作成イベントのフィールド 1 を活性化 153
4.28. ユーザー選択でのField1を活性化 154
4.29. Linux上でOTRSを開始するサンプル 197
4.30. OTRSスケジューラを開始するサンプル 198
4.31. OTRSスケジューラを強要停止するサンプル 198
4.32. WindowsサービスとしてOTRSスケジューラを登録するサンプル 199
4.33. ウィンドウズ上のOTRSスケジューラを開始するサンプル 199
4.34. Windows上でOTRSスケジューラを強制停止するサンプル 200
5.1. ACLはチケット優先度5のチケットのみをキューへの移動を許可します。 201
5.2. データベースに格納された優先度5のチケットだけのキューへ移動を許可するACL 201
5.3. 未対応のキュー中のチケットの完了を不能にし、閉じるボタンを隠すACL 202
5.4. 状態：「閉鎖成功」を常に削除するACL 203
5.5. "HW"で始まるキューが作成されるハードウェア・サービス・チケットのみを表示する
ACL 203
5.6. 全ての可能かつ重要なACL設定を示すリファレンス (参照) 203

序文

OTRS 管理者向けの冊子です。OTRS 初心者のための参考書としても提供しています。

次の章では、インストール、構成、および OTRS ソフトウェアの管理について説明します。テキストの最初の 3行で、ソフトウェアの主要な機能について説明しています。一方残りの部分は、構成可能なパラメタの完全なセットへの参照として提供されています。

この本は新製品中の移動標的を与えられて、引き続き進行中の仕事です。私達はこれを使いやすく正確かつ完璧な高品質な参考文献にするためにあなたのフィードバックを必要としています。この本で内容の欠落、説明が不十分な点、スペルミス、文法のエラーあるいは誤植を見つけた場合は私達にお知らせ下さい。どのような種類のフィールドバックも正しく評価され、 http://bugs.otrs.orgにある 私達の追跡システムによって反映されます。あなたの貢献に感謝申し上げます。

第1章 導入

1. トラブル・チケット・システム - 基礎

この章ではトラブル・チケットの主要概念の説明とともにトラブル・チケット・システムの手短な紹介を提供します。例示により手っ取り早くこのようなシステムを使うメリットについて説明します。

1.1. トラブル・チケット・システムとは何でしょうか？なぜあなたはそれを必要としているのでしょうか？

次の例ではトラブル・チケット・システムとは何か、そしてどのようにあなたが会社でそのようなシステムが役立つかを述べています。

マックス氏は、ビデオレコーダーの製造業者だと想像して下さい。マックス氏は製品サポートを必要とするお客様から多くのメッセージを受け取っています。ある日、彼は速やかに返答することもメッセージを受け入れることもできなくなりました。一部の顧客は対応の遅さに我慢ができず同じ質問をするために 2 度めのメッセージを送ります。サポート依頼を含むすべてのメッセージは、単一の受信トレイ フォルダーに格納されます。サポート依頼はソートされておらず、マックス氏は、通常のEメール プログラムを使用してメッセージに応答します。

マックス氏はすべてのメッセージに対して十分に速やかに返信できないため、彼は開発者のジョー氏とジョン氏に助けてもらっています。ジョー氏とジョン氏は同じメールシステムを使い、同じメールボックスにアクセスします。彼らはマックス氏がしばしば一人の不満を持った顧客から 2 件の全く同じ要求を受けていることに気づいていません。ときどき彼ら二人は全く同じサポート依頼に対して別々に返答をし、顧客は二つの異なる返答を受信してしまうはめになります。その上、マックス氏は彼らの返信の詳細について知りさえしないのです。また彼は、どの問題がより頻繁に起こっているいるのかといった顧客の問題やそれらの解決の詳細についても、あるいは、顧客サポートにどれほどの時間やお金が使われているかについても気づいてはいません。

会議にて同僚はマックス氏に対して、トラブル・チケット・システムが、如何にマックスが顧客サポートで抱えている問題を解決可能であるかを伝えました。インターネットで情報を調査したのち、マックス氏は、コンピュータにOTRSをインストールすることに決めました。それは彼の顧客と彼の従業員の両者がウェブからアクセスできるものです。現在では顧客のサポート要求はもうマックス氏の個人メールボックスに送られることはなくOTRS用に使用されるメールアカウントに送信されます。チケットシステムはこのメールボックスに接続しすべての顧客の要求をデータベースに保存します。新しい要求毎に、システムは自動的に解答を生成し、それを顧客に送ります。そのため顧客は要求が受信されて速やかに要求に対し解答されるだろうとわかります。OTRSは明確な参照番号、すなわち、チケット番号を生成し、要求毎に対して与えます。現在では顧客は喜んでいます。なぜなら顧客の要求は認識され、同じ内容の 2 度目の要求メッセージを送らなくても良くなったからです。マックス氏、ジョン氏そしてジョー氏は現在では、シンプルにウェブからOTRSにログインして顧客の要求に回答しています。システムは回答中のチケットをロックするので、そのメッセージを同時に他の誰かに編集されることはありません。

スミス氏がマックス・カンパニーにサポート要求をして、彼のメッセージがOTRSで処理されたと想像してみましょう。ジョン氏は彼の要求に短い解答をします。しかしスミス氏は彼の質問に対するジョンの解答に対して引き続き質問をしたいと思っています。ジョン氏は多忙のためマックスがスミス氏のメッセージに解答します。マックス氏は、OTRSの履歴機能によってこの要求に関するコミュニケーションの全ての流れを把握することができます。スミス氏は複数のサービス担当者が
彼の要求を解決するためにかかわっていることを知らないのですが、彼はマックス氏の最後の解答が到着しその詳細について大変満足しています。

むろん、これはトラブル・チケット・システムのほんのわずかの可能性と特徴を垣間見たに過ぎません。しかし、もしあなたの会社がEメールと電話を通して大量の顧客要求を処理しなければならないならば、そして、もし異なるサービス担当者が異なる時間に対応する必要があるならば、チケット・システムは大変役立つことができます。むろん、これはトラブル・チケット・システムのほんのわずかの可能性と特徴を垣間見たに過ぎません。しかし、あなたの会社がEメールと電話を

使って大量の顧客要求を処理しなければならないならば、そして、もし異なるサービス担当者が異なる時間に対応する必要があるならば、チケット・システムは大変役立ちます。またチケット・システムは、ワークフロー・プロセスを合理化する助けとなり、効率化を促進し、全体的に生産性を改善します。チケットシステムは柔軟にサポートやヘルプデスク環境を構築することができます。顧客と2人のサービススタッフ間のコミュニケーションはよりわかりやすくなります。最終的な成果は、サービスの効率性が向上することです。そしてきっとその顧客の満足度は、会社により良い財務実績をもたらすことでしょう。

1.2. トラブル チケットとは？

トラブル チケットは、病院の患者のために作成した医療レポートに似ています。患者が病院を訪れるとまず、すべての必要な個人と医療情報を保持するために医療レポートが作成されます。何度も通院すると、患者は前回と同じあるいは他の医師に診察され、診察医は、患者の健康及び治療中の処置に関する情報を追加しレポートを更新します。こうすることで他のどの医師や看護スタッフがすぐに患者の状態を完全に把握することができます。患者が回復し退院するとき、医療レポートから得られるすべての情報は保存されレポートは閉じられます。

OTRSのようなトラブル・チケット・システムはトラブル・チケットを通常のEメールのように取り扱います。メッセージはシステムに保存されます。顧客が依頼を送信すると、システムは新しいチケットを生成します。そのようなシステムは新しく作成される医療レポートに相当します。新しいチケットへの応答は、医者が医療レポートへの記入することに似ています。チケットは、回答が顧客へ返送された場合あるいは、チケットがシステムによって個別に閉じられた場合に、完了となります。もし顧客がすでに完了したチケットに再度応答すると、そのチケットは、新しい情報を追加された上で、再対応中となります。全てのチケットは格納され、完全な情報でアーカイブされます。チケットは通常のEメールのように取り扱われ、添付ファイルや文脈上の注釈もまた各Eメールに格納されます。さらに、基準日、従業員関係、チケットの解決に要した作業時間などの情報が保存されます。その後いつでも、チケットはソートされ、検索可能であり、さまざまなフィルタリング・メカニズムを使いすべての情報を分析することができます。

2. OTRS ヘルプデスク

この章では、OTRS ヘルプデスク (OTRS) の機能について説明します。OTRSのハードウェアとソフトウェアの要件に関する情報を提供します。さらにこの章では、コミュニティとの連絡方法、そしてもし必要ならOTRSに関する商用サポートを取得する方法を知ることができます。

2.1. 基礎

OTRS ヘルプデスク (OTRS) は、 ウェブ サーバーにインストールされるウェブアプリケーションです。したがって　ウェブ ブラウザーで使用することができます。

OTRS ヘルプデスク (OTRS) は、 ウェブ サーバーにインストールされるウェブアプリケーションです。したがって　ウェブ　ブラウザーで使用することができます。OTRS　はいくつかのコンポーネントに分かれています。主なコンポーネントは、アプリケーションとチケット システムのすべての主要機能を含んでいる OTRS フレームワークです。以下のような追加のアプリケーションをインストールすることが可能です。OTRS::ITSM モジュール、ネットワーク監視ソリューションとの統合、知識ベース (サポート技術情報　) (FAQ) など。

2.2. 機能

OTRSには多くの機能があります。OTRS フレームワークに含まれる主な機能の概要を以下に示します。

2.2.1. ユーザ・インターフェイス

- OTRS には、担当者と顧客用に別々の最新のウェブ インターフェイスが付属します。
- モバイルプラットフォーム、ラティナディスプレ対応、どのモダンブラウザでも利用可能です

- ウェブインターフェイスは、独自のテーマとスキンでカスタマイズできます。
- 強力でカスタマイズ可能な担当者用ダッシュボードは、個人チケット概要とグラフィカルな統計情報のサポートを実装しています。
- 拡張可能なレポート作成エンジンは、様々な統計とレポートのスケジュール・オプションを提供します。
- プロセス管理でチケット・ベースの独自の画面とプロセス (チケット・ワークフロー) を定義することが可能です。
- OTRS は、権限管理システムを内蔵しており、きめ細かいアクセス制御リスト (ACLs) で拡張することができます。
- 30 以上の言語と異なるタイムゾーンをサポートします。

2.2.2. メール・インターフェース

- MIME メール添付ファイル付きのメールをサポートします。
- HTML形式メールをテキスト形式に自動変換します(機密性の高いコンテンツのセキュリティを強化し検索をより高速にします)。
- 受信メールはフィルタリングされ、複雑な規則のある、例えばスパムメッセージまたはキュー配布に対するなどの前処理が可能です。
- キー証明書の管理とメールの処理のための PGP鍵 と S/MIME 準拠のサポート。
- 全てのキューに設定可能な自動応答。
- 新規、フォローアップあるいはロック解除されたチケットに関する担当者へのEメール通知。
- フォローアップ (たとえばコール#、チケット#、あるいはリクエスト# など) を認識するために独自のチケットの識別子を定義することが可能です。さまざまなチケット番号生成器 (日付ベース、ランダムなど) があり、あなたがあなた自身をもそれに統合することができます。フォローアップもまたIn-Reference-To　headers (メール参照ヘッダー) または外部チケット番号で認識できます。

2.2.3. チケット

- OTRSは、外部、内部に所属している全てのコミュニケーションを収集するためにチケットを用います。これらのチケットは、キューで整理されます。
- 異なる詳細なレベル(小/中/プレビュー)においてシステム内のチケット (キュー、状態、エスカレーションなど) を閲覧するための多くそして異なる方法が用意されています。
- チケットの履歴は、チケットのあらゆる変更を記録します。
- チケットは、さまざまな方法で変更されます。返信、転送、bouncing、別のキューへの移動、属性 (状態、優先度など) の更新、ロック、稼働時間の計算という方法です。チケットの一括変更も可能です。 (bulk action)
- 保留中時間とエスカレーション時間 /SLA 管理は、チケットに関する時間ベースのスケジュール設定と制限を許可します。
- チケットは他のチケットまたは FAQ エントリなどの他のオブジェクトにリンクできます。
- 自動およびタイムアクションのチケットが 「GenericAgent」で利用可能です。
- OTRSは、チケットに関する複雑でフルテキストの検索が可能な強力なサーチエンジンとなります。

2.2.4. システム

- OTRSは、多くのオペレーティング・システム(Linux、Solaris、AIX、FreeBSD、OpenBSD、Mac OS 10.x、Microsoft Windows)上で走り、主要なOTRSバックエンド用のいくつかのデータベース・システム(MySQL、PostgreSQL、オラクル、MSSQL)をサポートします。
- コア・システムは、OTRSパッケージのインストールにより拡張することができます。OTRSグループのサービス契約顧客が利用可能な機能追加パッケージと同様に、多くの無料のパッケージも(FAQ、OTRS::ITSM他のような)あります。
- 顧客情報用の外部バックエンドの統合（例えば AD、eDirectoryあるいはOpenLDAP経由）。顧客は、データベース、LDAP、HTTPAuth あるいは Radius によって認証されます。
- 一般インターフェイス（GenericInterface）で、他のウェブサービスにOTRSを接続するのは簡単です。シンプルなウェブサービスによって、プログラムやカスタム拡張による複雑なシナリオなど必要とせずに、統合することができます。OTRSチケット・コネクターは、サードパーティー・アプリケーションのウェブサービスによって、チケットの更新、検索および作成を許可します。

では、次にOTRSの最近のバージョンの変化を見ましょう。

2.2.5. OTRS 3.3の新たな機能

2.2.5.1. 生産性

- ダッシュボード・チケット・リストおよび通常のチケット概要では、適格なチケット列をフィルターし、表示列を構成可能になりました。
- チケット・ミディアムおよびプレビュー概要が、ソート可能になりました。
- チケットをイベントとして表示することができるダッシュボード用のカレンダー・ウィジェットを追加しました。
- 状態値およびキュー当たりのチケットの数から、表形式で表示する新しいダッシュボード・ウィジェットの追加。
- 担当者は重要な記事をマークすることができるようになりました。
- 新しいツリー選択ウィジェットは、ツリーデータ（キュー、サービスなど）を用いて作業をより速く、より簡単にします。
- 曜日および日付時刻といった動的なフィールドで、相対日付(例えば1か月以上前に)を検索するサポート機能を追加。
- テンプレート(以前は「標準レスポンス」)を指定することができます。新しいチケットの生成および転送のためにも使えます。
- 利用可能なプロセスのリストは、ACLによってフィルターすることができるようになりました。
- カスタマー・インターフェースからのプロセスを始める追加サポート。
- 多くの場合テキストは文字の定数(「Queue1...」...)によって短くはできませんが、利用可能な画面（解像度）によって可能になります。それによって一覧でより多くの情報を見ることができます。
- OTRSがレティナ・ディスプレイ対応となりました。イメージはより高い解像度に適応されています。また、ほとんどのイメージ・アイコンは FontAwesome webfont からのフォント文字と取り替えられました。
- 新機能「管理ダッシュボード」が追加され、ダッシュボード中で統計の図表を表示できるようになりました。ただし、IE8はこの機能をサポートしておりませんのでご留意下さい。

2.2.5.2. 外部システムとの作業

- OTRSは、複数の顧客会社データベースを使用することができるようになりました。（Cyrille@belnet-ictに感謝！）
- OTRSは、顧客ユーザーデータをチケットの動的フィールドに（チケットに永久保存するため）格納するようになりました。これはレポーティングに役立つ場合があります。
- OTRSは、外部システムから受信したEメールを、チケット番号に基づいて既存のチケットへ、正しく割り当てることができるようになりました。
- OTRSは、POP3/TLS接続経由でEメールを取得(fetch)することができるようになりました。

2.2.5.3. インストールとアドミニストレーション（管理）

- ウェブインストーラーは、MySQLに加えてのPostgreSQL、オラクルおよびSQLサーバー・データベースの上でOTRSをセットアップできるようになりました。
- OTRSは、MySQL 5.6をフルサポートするようになりました。
- 一般の担当者ジョブが、設定されたチケット・イベントのために実行することができるようになりました。
- 新しいグラフィカルなACLエディターはACL編集をより簡単にします。
- ポストマスター・フィルターによっては、無効にされたフィルター条件を使用することができるようになりました。（Renee Backerに感謝！）
- ポストマスター・フィルターは、関連した滞留日と所有者あるいは、受信するEメールデータに基づく新しいチケットに対する責任者を特定するようになりました。
- 顧客と担当者のパスワードは強力なbcryptアルゴリズムを使って暗号化することができるようになりました。bcryptはSHAより良質です。
- 多くのアイコンはアイコン・フォントを使うことができるようになりました。それによりさまざまな基調色でカスタム・スキンを作成することが容易になりました。

2.2.6. OTRS 3.2の新機能

2.2.6.1. 顧客をより重視すること

- 新しい "顧客情報センター" は、顧客（会社）に大きなダッシュボードのような概要を提供します。
  - 顧客の会社のチケットを、エスカレーション、リマインダー、新規作成、そして対応中にします。
  - 顧客ユーザ（連絡先）は、この顧客会社に所属しており、それら個別のチケットカウントと新規チケットを作成するためのショートカットを持っています。
  - 顧客会社の総合的なチケット状態のビュー
- 新しい「顧客へのスイッチ」の機能によって、担当者は、必要な許可さえあればその権限に応じて、顧客のパネルを見ることができるようになりました。

2.2.6.2. さらなるカスタマイズが可能

2.2.6.2.1. 工程管理

- 新しい工程管理 process management は、OTRSの内のプロセスを表わすことを可能にします。

2.2.6.2.2. 顧客インターフェースの改良

- 顧客ウェブインターフェースは全面的にAJAXとACLをサポートするようになりました。
- 顧客ウェブインターフェースは、JavaScriptを必要とし、IE6以前バージョンと互換性がなくなりました。
- 顧客インターフェースでは、新しいチケット用にデフォルト・チケット・タイプをセットすることができるようになりました。さらにチケット・タイプを非表示とし、デフォルト値を顧客インターフェース経由で作成されたすべてのチケットに使用することができるようになりました。

2.2.6.2.3. 担当者インターフェースの改良

- 担当者は、エスカレーション時間に基づいてチケットを検索することができるようになりました。
- チケット検索のデフォルトで DynamicFields を表示する新しいオプション。
- ポップアップ・ウィンドウ中でスクロールしないようにするためのチケット画面中の画面利用最適化。各チケット画面については、リッチ・テキスト・エディタのサイズは個別に設定できるようになりました。
- 構成オプションを活性化した後に、TicketAction　ダイアログ (TicketNote、TicketCloseなど) の内部からの別のキューへの移動が可能になりました。この機能はデフォルトではオフです。
- チケットが 1 枚しかない場合、チケット検索からチケット・ズーム画面に直接ジャンプするようになりました。
- ブラウザ・ウィンドウに、より多くのデータを調整表示するために役立つ TicketActionCommon ベースの画面から記事のタイプを非表示にする新しい機能。
- 現在利用不可能な担当者をすべてリストする、新しいオフィス不在・ダッシュボード・ウィジェットがあります。
- 新しい CKEditor 4は、リッチ・テキスト内容(HTML Eメールのような)での作業をより簡単で安定させます。

2.2.6.2.4. 管理の改善

- イベント・ベースの通知は、特定の記事送信者タイプのためにのみ送信することができます。
- OTRSの中の統計エンジンは、日次、月次、年次にに加えて「週次」を理解するようになりました。これにより、例えば、「先週作成されたチケットのための報告書を作成します。」あるいは1週当たりの1つのキュー当たり作成されたチケットを表示する報告書を生成する、といったことが可能になりました。
- `Custom/Kernel/Output/HTML`　にカスタマイズしたDTL(テンプレート)ファイルを置くことは可能です。そのため、それらはすでにPerlファイルでは稼働しているように、システムのデフォルトのDTLファイルを無効化(override)します。
- AdminSMIME では、人が読める証明書コンテンツを表示することができます。
- SysConfig は、Types Date と DateTime を設定をサポートするようになりました。

2.2.6.3. 拡張性の向上

2.2.6.3.1. 改善されたチケットのアーカイブ

- チケットがアーカイブに保管される際、担当者のチケット購読と同様に、担当者が読んだチケットや記事の情報は、削除されます。これはデフォルトで有効で、それにより、多数のチケットや担当者を伴う大規模システムのデータベースのデータ量を減らすことに役立ちます。

- さらに既存のアーカイブに保管されたチケットからこのデータを取り除く、新しいスクリプトがあります。
- アーカイブに保管されたチケットは、担当者には「read」として常に表示されます。

2.2.6.3.2. パフォーマンスの改善

- セッション管理は特に多くの活動的なユーザにおいて、10倍近くまで高速化されました。
- サーバー能力過負荷を回避するために担当者および(または)ユーザの数を制限することができるようになりました。
- 担当者および顧客フロント・エンドの中のチケット概要およびチケット・マスク中の実行されたデータベース命令文数の著しい減少。
  - これは、特に大規模システム上で、データベース・サーバ上のロードを軽減するでしょう。ある場合には、OTRSが、明白に、もっと反応が良くなるでしょう(システムがDBロードか遅延によって遅くなった場合)。
- LDAP ユーザ同期のパフォーマンスの改善。
- 多くのキャッシュ・ファイルでキャッシュ・パフォーマンスの改善。

2.2.6.4. 相互運用可能性の改善

2.2.6.4.1. GenericInterface（一般インターフェイス）のための**FAQ**コネクター

- ウェブサービス（GenericInterface）経由で　FAQモジュール（OTRS知識データベース）のデータにアクセスすることができるようになりました。例えば、これはあなたの会社ウェブサイト上のFAQ記事を埋め込むのに役立ちます。

2.2.7. OTRS 3.1の新しい機能

2.2.7.1. 一般インターフェース - ウェブサービス・フレームワーク

- GI は、サードパーティー・アプリケーションを持ち、OTRS のウェブサービス相互接続が可能で柔軟なフレームワークです。
- OTRSは、両方の方法で操作可能です　-　供給者（サーバ、遠隔操作要求者）あるいは、依頼人（遠隔操作要求する顧客）として。
- 一般インターフェースの設定によりプログラムせずにシンプルなウェブサービス接続を作成することができます。
- さまざまなアーキテクチャ層のGIインフラストラクチャに　Perlコードを追加するカスタムOTRS拡張をプラグインすることによって複雑なシナリオも実現することができます。
- コネクターは、一般インターフェース・ウェブサービスに、OTRSの部分を露出し（向け）ます。たとえば、チケット・コネクターはチケット作成・更新機能に露出し（向け）、その結果、それらはウェブサービスで使用することができます。ウェブサービスがどのネットワークを使用しているかに無関係に。
- スケジューラ・デーモン・プロセスは非同期イベントの取り扱いをサポートします。これは、OTRSから他のシステムまでウェブサービスリクエストを非同期にスタートするために有用です。担当者のリクエストが返答された後で（例えば例えばチケットが作成された時）。そうでなければ、それは担当者の対応時間の増加につながり、対応をブロックするかもしれません。

一般インターフェースで、新しいウェブサービスは、補足コードなしで既存のOTRSモジュールの使用により容易に設定することができます。それらは新しいウェブサービスを作るために組み合わせることができます。新しいウェブサービス接続を設定する場合、管理者は次のことを追加しなければなりません:

- 管理者用GUIの新しいウェブサービス
- 基本的なメタデータ (トランスポート・タイプ (SOAP) 、URL等) と
- 既存のオペレーション(コネクターの一部)、そして各オペレーションのためにデータをマップ(受信と送信を)される方法をを明示特定します。

一般インターフェース・デバッガは、OTRS管理者が、リクエストがどのように中へ入るか、また、それらがどのように異なるレイヤーに扱われるかを、チェックするのを助けるでしょう。

2.2.7.1.1. 現在の機能

- ネットワーク・トランスポート： SOAP/HTTP、その他のREST やJSONが、顧客からの要求次第で将来追加される予定です。
- 受信・送信データに関しての　キー/値変換用の設定可能なデータマッピング・グラフィカルユーザーインターフェイス。
- 設定されたウェブサービスの情報のフローと設定をチェックするグラフィカルなデバッガ。
- チケット取り扱いのためのウェブサービスとしてOTRSの使用を可能にするチケット・コネクター。

2.2.7.1.2. 将来の機能

- 追加のネットワーク・トランスポート(REST、JSON)。
- GI　はモバイル用アプリケーションのバックエンドとして　iPhoneHandle　に代替されるでしょう。
- 追加のコネクタは、ウェブサービスを使用するOTRSのより多くの部分を提供するために追加されるでしょう。 (例えば、担当者、ユーザ、サービスあるいはCIの作成、更新、削除を許可すること)

2.2.7.2. 動的フィールド

DynamicFields　の機能は、OTRSでユーザー設定のフォームを作成することも可能にする動的構造と、既存のチケットおよび記事 FreeText、および FreeTime フィールドとを代替します。

- 自身の管理者用GUIを使って無制限の量のフィールドを設定することができます。
- フィールドにはチケットおよび記事の両方のために使用することができる、異なるタイプが存在します。デフォルトで利用可能なものは以下の通りです:
  - テキスト
  - 複数行テキスト
  - チェックボックス
  - ドロップダウン
  - 複数選択
  - 日付
  - 日時
- 新しいカスタム・フィールド・タイプ　(例えば外部データ・ソースを伴うカスタム・フィールド・タイプ：ドロップダウン) は、プラグイン接続できるモジュラーで作成されるフィールドとして少ない労力で追加できます。

- 将来のシナリオは以下の通りです。DynamicFieldsは、チケット以外の、あるいはカスタム・モジュール中のオブジェクトに使用することができます。例えばOTRSでは「注文 (order)」を取り扱うオブジェクトを追加するカスタム・モジュールは、これらの「注文」にプロパティ/データ（properties/data）を付加するためにDynamicFields動的フィールドを使うことができます。
- データベース更新スクリプトは、時系列のFreeTextフィールドおよび関連設定を新しい構造に変換します。

2.2.7.3. チケット管理の改善点

2.2.7.3.1. 改善されたチケット生成

- 新規の電話やEメールアドレスを作成する時に、複数のEメールアドレスを'To:', 'CC:' or 'BCC:'として指定できます。

2.2.7.3.2. 着信呼のサポート

- 着信呼が既存のチケット内に（現在のところ発信呼のみ登録されています）登録されるようになりました。

2.2.7.3.3. 改善されたチケット概要のプレビュー

- 特定の送信者のタイプ（例えば内部担当者の記事）の記事を除外することが可能となりました。概要プレビューモードのSysConfig内で表示することができます。
- 閲覧時、特定の記事のタイプ（それはデフォルトで拡張されたタイプ記事を表示します）を設定可能です。

2.2.7.3.4. 改善されたチケットの移動

- チケットを移動させた後に表示される画面はが設定可能になりました。そのオプションは、チケット・ズーム・ビュー(LastScreenView)あるいはチケット・リスト(LastScreenOverview)です。

2.2.7.3.5. バルク・アクション（Bulk action）の改善

- 新規のバルク・アクションにて、同時に複数のチケットからEメールを発信可能になりました。チケットは異なるキューをもち、これらのキューは各々異なるテンプレート、挨拶文、署名を持ちます。これらはBulk Action Eメールでは使用されません。
- 追加のバルク・アクションは選択したチケットに対してチケット・タイプを設定することを可能にします。

2.2.7.3.6. 構成可能な受信拒否するEメール・アドレス

- この機能は、メールによって新規チケットの作成を拒否するために administrator アドレスの代わりとなるEメールアドレスを構成することを許可します。この機能は、顧客がメールで新規チケットを作成できないような全てのケースで使われます。

2.2.7.4. プロセス・オートメーション

2.2.7.4.1. 追加されたエスカレーション・イベント

- OTRSは、利用可能なエスカレーション・タイプ(返答、更新あるいは解決)を作成することができるようになりました。この機能は、エスカレーションが発生する前後、あるいは発生しているまさにその瞬間に行動（たとえば通知）をとることを可能にします。

2.2.7.4.2. 改善された通知メカニズム

- 新しい一般担当者向け通知モジュールは　OTRS管理者がメッセージを定義できるようにしました。そのメッセージは、担当者がシステムにログインする時に担当者のウェブフロントエンドに表示されます。

2.2.7.4.3. 時間計算の改善

- すべての時間類は、アプリケーション・サーバーによってあるいは基づいて計算されます。アプリケーションとデータ間の時計の時刻の変数に起因する問題を解決するためです。

2.2.7.4.4. 改善されたGenericAgent (一般担当者)

- GenericAgentはチケットの変更時間に対してフィルターできるようになりました。
- さらに、GenericAgentは一致したチケットの責任者を設定可能になりました。

2.2.7.5. ユーザ・インターフェース、リッチ・テキストエディター、CHARSET (文字セット)

2.2.7.5.1. 改善されたユーザ・インターフェース・パフォーマンス

- レンダリングのスピードと記事表示速度が改善されました。(Stelios Gikas<stelios.gikas@noris.net>に感謝！)

2.2.7.5.2. リッチ・テキストエディターのアップデート

- IOS5 のサポートを追加
- ブロック引用はエンター・キーで終了します。
- CKEditorのバージョンは3.4から3.6にアップデートされ、その改善はCKEditor 3.5およびCKEditor 3.6のリリースを参照下さい。
- IE9サポートを改善しました。
- サイズ変更可能なダイアログ。

2.2.7.5.3. **Unicode**サポート - 内部エンコードとして**UTF-8**以外を非サポート

- UTF-8はOTRSの許可された内部唯一のキャラクタセットとなりました。
- 言語ファイルはすべてUTF-8にフォーマットされるようになりました。それは、言語ファイルの取扱いと将来的な翻訳メカニズムの改善を簡略化します。

2.2.7.6. データベース・ドライバーのサポート

2.2.7.6.1. 改善された**PostgreSQL**ドライバー互換性

- PostgreSQL 9.1 のサポート追加。
- レガシーのドライバーは、PostgreSQL 8.1以前のバージョンで利用可能になりました。

2.2.7.6.2. 改善された**Ms SQL**ドライバー互換性

- MS SQL ドライバーは、あまり使われなくなったタイプのTEXTよりも、バイナリデータをVARBINARYに格納します。また、同様に、テキスト文字列を保存するためにVARCHARよりもNVARCHARに格納します。(改善されたユニコードのサポートのため)

2.2.7.7. **E**メールの統合

2.2.7.7.1. 改善された**E**メール取り扱い

- IMAP Eメールアカウントに接続時、INBOXフォルダーに加えて特定のEメール・フォルダーからのEメールを扱うことができるようになりました。
- OTRSは、トランスポート層セキュリティ(TLS)を使用して最新の制約された環境でIMAPサーバに接続できます。

2.3. ハードウェアとソフトウェアの必要条件

OTRSは、様々なオペレーティング・システムにインストールすることができます。OTRSはlinuxおよび他のunix派生的(例えばOpenBSDまたはFreeBSD)の上で走らせることができます。さらに、マイクロソフト・ウィンドウズ上でそれを実行することができます。OTRSには過度のハードウェア要件がありません。最少構成のセットアップのためには、最低Xeon - 2GHz またはその互換CPU、2GB RAM, 160GB HDDのマシンをご利用されることをお勧めします。

さらに、OTRSを実行するために、ウェブサーバとデータベース・サーバを使用する必要があるでしょう。それとは別に、perlをインストールし、かつ、またはいくつかの追加のperlモジュールをOTRSマシンにインストールするべきです。OTRSと同じマシンにウェブサーバとパールをインストールしなければいけません。データベース・バックエンドは、ローカルにあるいは別のホスト上でインストールしても結構です。

ウェブサーバについては、Apache HTTP Serverの利用をお勧めします。そのモジュールmod_perlがOTRSの性能を非常に改善するためです。それとは別に、OTRSは、Perlスクリプトを実行することができるウェブサーバ上で走らせるべきです。

異なるデータベース上のOTRSを展開させることができます。MySQL、PostgreSQL、OracleあるいはMicrosoft SQL Serverのいずれかを選択可能です。MySQLを使用すれば、ウェブ・フロントエンドによって、インストール中にデータベースおよびいくつかのシステム・セッティングを設定することができるという長所を持ちます。

Perlは最低バージョン5.8.8を使用することを推奨します。Perl・シェルおよびCPANで、あるいはオペレーティング・システム(rpmおよびyast、apt-get)のパッケージ・マネージャーによってインストールすることができるいくつかの追加モジュールを必要とするでしょう。

ソフトウェア必要条件

2.3.1. Perlサポート

- Perl 5.10以上

2.3.2. ウェブサーバ・サポート

- Apache2 + mod_perl2、またはそれ以上(推奨)
- CGIサポート(CGIは非推奨)を備えたウェブサーバー
- マイクロソフト・インターネット・インフォメーション・サーバー(IIS)6以上

2.3.3. データベース・サポート

- MySQL 5.0以上
- PostgreSQL 8.0以上(8.2以上推薦)
- Oracle 10g以上
- Microsoft SQL Server 2005以上

Perl・モジュールのインストールに関するマニュアル中のセクションは、OTRSに必要とされるものをどのようにセット・アップすることができるかをより詳細に説明しています。

あなたがOTRS(それらはあなたのオペレーティング・システム(rpm、Windowsインストーラー)のために構成された)のバイナリ・パッケージをインストールする場合、パッケージはPerl・モジュールをすべて含んでいるか、あるいは、システムのパッケージ・マネージャーは、Perl・モジュールの依存性に留意する必要があります。

2.3.4. ウェブブラウザー・サポート

OTRSを使用するために、JavaScriptをサポートする最新のブラウザを使用すれば、問題ありません。次のブラウザはサポートしません:

- Internet Explorer バージョン 8 以前
- Firefox バージョン 10 以前
- Safari バージョン 5 以前

ブラウザの最新バージョンを常に使用することを推奨します。それは最良のJavaScriptとレンダリングパフォーマンスを有するためです。巨大なデータあるいは巨大なシステムでブラウザを使うとそのパフォーマンスに劇的な違いが生じます。その問題に関してご意見頂ければうれしく思います。

2.4. コミュニティー

OTRSには大きなユーザ・コミュニティーがあります。ユーザと開発者はメーリング・リストによってOTRSおよび関連銘柄についての交換情報について議論します。OTRSのインストール、構成、使用法、各国言語化および開発について議論するためにメーリング・リストを使用することができます。私たちのバグ・トラッキング・システム中のソフトウェア・バグを報告することができます。

OTRSコミュニティーのホームページは次のとおりです:http://www.otrs.com/open-source/ .

2.5. OTRSのためのプロフェッショナル・サービス

OTRSを設定するかカスタマイズする際にヘルプを必要とするか、あなたが安全に運用したいかにかかわらず、遠慮なく、私たちと連絡をとってください:　私たちは広範囲の世界的な企業サポートのようなプロフェッショナル・サービス提供します。それは、工程設計、インプリメンテーション、カスタマイズ、アプリケーションサポートおよび完全に管理されたサービスを含むコンサルティングやエンジニアリングです。

私たちのサービス契約 は、サポート評価と同様に、即時のヘルプおよび専門家のサポートも保証します。そしてOTRS機能アドオンへの最新のしかし可能な限り無料のアクセス- あなたのOTRSのための有用な追加の機能です。

OTRSグループ は、様々な国々の特定のトレーニング・プログラムを提供します。定期的に開催する私たちの公のOTRS管理者トレーニングのうちの1つに参加するか、あるいは社内の特定の要求をすべてカバーするための社内トレーニングから利益を享受することもできます。

第2章 インストレーション

本章は、主要なOTRSフレームワークのインストレーションおよび基礎的な構成について記述します。OTRSをインストールする情報を、ソース、あるいはRPMまたはウィンドウズ実行可能なバイナリ・パッケージでカバーします。

ここでカバーするトピックは以下を含みます。ウェブおよびデータベース・サーバ、OTRSとデータベースの間のインターフェース、追加のパール・モジュールのインストール、OTRSのためのセットする適切なアクセス権、OTRSのためにcronジョブをセット・アップすること、OTRS設定ファイル中のいくつかの基礎的な設定の構成、です。

本章の詳細なステップに従ってOTRSをサーバにインストールしましょう。インストールが完了したらウェブ・インターフェースを使用してシステムにログインし管理することができます。

1. シンプルな方法 - あらかじめ構築されたパッケージのインストレーション

ご利用のプラットフォームで利用可能な場合、OTRSをインストールするためにあらかじめ構築されたパッケージを使用するべきです。それが最も簡便な方法ですから。ダウンロード・エリアhttp://www.otrs.comでそれらを見つけることができます。次のセクションは、SUSE、Red Hatおよびマイクロソフト・ウィンドウズ・システムの上のあらかじめ構築された、あるいは、バイナリ・パッケージによるOTRSのインストールについて記述します。なんらかの理由であらかじめ構築されたパッケージを使用することができない場合のみ、手動のプロセスに従うべきです。

1.1. SUSE Linuxサーバ上でのRPMのインストール

このセクションは、SUSEリナックス・サーバーの私たちのRPMのパッケージの設置について記述します。私たちはすべての最近のSLESおよびopenSUSEバージョンに対してテストしました。インストレーションを始める前にhttp://www.otrs.com/downloadsを訪ね、利用可能な最新のOTRS RPMのパッケージを使用しているかをご確認下さい。

1.1.1. OTRS用データベースの準備

異なるデータベース・バックエンドを使用して、OTRSを使用することができます:MySQL、PostgreSQL、オラクルあるいはMicrosoft SQL Server。OTRS上に展開させる最もポピュラーなデータベースはMySQLです。本章は、SUSEベースのサーバ上のMySQLを形成するために取る必要のあるステップを示します。もちろん、もし拡張性あるいは他の目的で必要ならば、データベースを専用データベース・サーバにインストールすることができます。

> **注記**
>
> openSUSE 12.3以上に関する本章に従って、実際にMySQLではなくMariaDB（MySQLコードのMySQL互換フォーク）を代わりにインストールする場合。これは問題ではありません。かえって都合がよく作動するでしょう。(しかもいくつかの点で多少より良いです)

rootとして次のコマンドを実行しMySQLをインストールします：

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

デフォルト・オプションが付属したMySQLをシステムにインストールします。それをOTRSに適するようにデフォルトを変更する必要があるでしょう。テキストエディターで、ファイル/etc/my.cnfを開き、コマンド**max_allowed_packet**で行を変更し、以下のように一行追加しまう:

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

まず、データベース・サーバを再起動し、かつこれらの変更を活性化するために**rcmysql restart**を実行します。次に、**/usr/bin/mysql_secure_installation**を実行して、データベースrootパスワードをセットし、匿名のアクセスを削除し、テスト・データベースを削除し、スクリーン上の指示に従ってください。最後に、mysqlがサーバ起動時に自動的に開始されることを確かめるために**chkconfig -a mysql**を実行してください。

1.1.2. OTRSのインストール

コマンド**zypper**を使用するコマンドラインでOTRSをインストールします。これはApacheのウェブサーバおよびいくつかのパール・モジュールのようないくつかの依存性も誘起します。カレント・ディレクトリにOTRS RPMのファイルをコピーしたことを確かめてください。

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

Next steps:

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

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

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

             stop-force}).

((enjoy))

 Your OTRS Team
 http://otrs.org/

otrs-sles:~ #
```

図：*OTRS*をインストールするコマンド

OTRS設置は完了しました。以下のスクリプト中で示されるように、その構成中のOTRS特有の変更をロードするためにウェブサーバを開始してください。さらにサーバのリブート時OTRSが自動的に開始されることを確認するために**chkconfig**を実行してください。

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


    done
otrs-sles:~ #
```

スクリプト：ウェブサーバの開始

1.1.3. 追加perlモジュールのインストレーション

OTRSは、RPMによってインストールすることができるモジュールよりさらに幾つか多くのモジュールを必要とします。それらを手動で後からインストールすることができます。/opt/

otrsディレクトリーにあるbin/otrs.CheckModules.pl スクリプトを実行して、不足しているのがどのモジュールかをチェックすることができます。いくつかのモジュールは、IMAP(S)サーバーかPDF生成とのコミュニケーションのようなオプションの機能性にだけ必要とされます。SLESではシステムにモジュールを必要とするために外部リポジトリをzypper構成に追加すべきです。OSバージョンに必要とされたモジュールをここから選択してください:http://download.opensuse.org/repositories/devel:/languages:/perl/。SLE 11 SP2用にこのようにリポジトリを追加します。

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

openSUSE 12.3では、別のリポジトリがMail::IMAPClientモジュールにだけ必要とされます。TLSで安全なIMAPサーバーからメールを集める必要があるときだけ、それを必要とするでしょう。対応する行は以下ご欄ください:

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

このリポジトリ追加後、はじめてzypperを使用時、そのキーを追加するように促されるでしょう。これで以下のように、不足しているモジュールをインストールすることができます。

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

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

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

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

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

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

次のステップではこのセクションで記述されるように、ウェブ・インストーラーを使用して、OTRSを構成します。

1.2. Red Hat Enterprise Linux あるいは CentOS システムへのOTRSのインストール

このセクションは、Red Hat Enterprise Linux (RHEL)あるいはCentOSサーバー上のRPMのパッケージの構成について記述します。OTRS 3.3以上については、RHEL 6あるいはCentOS 6が必須です、バージョン5はサポートされません。インストールを始める前に、http://

www.otrs.com/downloadsを訪れ、利用可能な最新のOTRS RPMのパッケージを使用しているかを確認してください。

1.2.1. OTRS用データベースの準備

異なるデータベースバックエンドを使用して、OTRSを使用することができます:MySQL、PostgreSQL、OracleあるいはMicrosoft SQL Server。OTRSを上に展開させる最もポピュラーなデータベースはMySQLです。本章は、RHELベースのサーバ上のMySQLを構成するために必要なステップを示します。もちろん、拡張性あるいは他の目的に必要ならばデータベースを専用データベース・サーバにインストールすることもできます。

rootとして次のコマンドを実行しMySQLをインストールします：

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

これは、デフォルト・オプションを持つMySQLをシステムにインストールします。それをOTRSに適しているようにデフォルトを変更する必要があります。テキストエディターで、ファイル/etc/my.cnfを開き、[mysqld]セクションの下の次の2行を追加します:

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

ここで、データベース・サーバを再起動し、かつこれらの変更を活性化するために**service mysqld start**を実行します。次に、**/usr/bin/mysql_secure_installation**を実行して、データベースrootパスワードをセットし、匿名のアクセスを削除し、テスト・データベースを削除し、スクリーン上の指示に従ってください。最後に、mysqlがサーバ開始時に自動開始されることを確認するために**chkconfig mysqld on**を実行してください。

1.2.2. OTRSのインストール

コマンド**yum**を使用するコマンドラインからOTRSをインストールします。これは、さらにApacheウェブサーバおよびいくつかのPerlモジュールのようないくつかの依存性を誘起するでしょう。カレント・ディレクトリにOTRS RPMのファイルをコピーしたことを確認してください。

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

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

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

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

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

Next steps:

[httpd services]
 Restart httpd 'service httpd restart'

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

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

((enjoy))

 Your OTRS Team

Installed:
  otrs.noarch 0:3.3.3-01

Dependency Installed:
  ...

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

図：*OTRS*をインストールするコマンド

OTRS設置は完了しました。ここでApacheが開始し、サーバがリブートする場合は常に、Apacheが開始することを確認すべきです。

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

スクリプト：ウェブサーバの開始

1.2.3. 追加perlモジュールのインストレーション

OTRSは、RPMによってインストールすることができるよりさらに幾つかのモジュールを必要とします。それらを手動で後でインストールすることができます。/opt/otrsディレクトリーにあるbin/otrs.CheckModules.plスクリプトを実行して、不足しているのがどのモジュールかをチェックすることができます。いくつかのモジュールは、IMAP(S)サーバーかPDF生成とのコミュニケーションのようなオプションの機能性にのみ必要とされます。Red HatまたはCentOSの上でEPELリポジトリからのこれらのモジュールのインストールを推奨します。そのリポジトリは、RHELとそのは派生のための高品質パッケージを供給しているFedora projectによって維持管理されています。より詳細な情報は以下を御覧ください。the EPELウェブサイト。

RHEL 6あるいはCentOS 6をご利用の場合、EPELのために最新のパッケージを得ることができます。このサイトから。このページで見つけたRPM URLをコピーして、このコマンドを実行することによって、このリポジトリをyumに追加できます：

```
 [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]#
```

このリポジトリを追加後、yumの初回使用時、そのキーを加えるように促されます。ここで不足しているモジュールを以下のようにインストールすることができます。

```
[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]#
```

次のステップではこのセクションで記述されるように、ウェブ・インストーラーを使用して、OTRSを構成します。

1.2.4. Red HatまたはCentOSでのOracleデータベース・ドライバのインストレーション

Oracleデータベース上のOTRSを展開させたければ、DBD::Oracleデータベース・ドライバをコンパイルし、インストールする必要があります。これは他のどのパッケージのインストールよりわずかに複雑です;これはオラクルが所有権のあるデータベースであり、Red HatもCentOSもRPMレポジトリにドライバを配布することが許可されないプロジェクトだからという理由だからです。

まず、ドライバーをコンパイルしインストールできるように、gcc, make 及び CPANをインストールする必要があります。以下、CentOSに関するコマンドを見てみましょう;他のバージョンにおいては、それが少し異なっているように見えるかもしれません。

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

次のステップは、データベース・クライアントを入手しインストールすることです。これについては、オラクルのウェブサイトで無料アカウントのサインアップをする必要があるでしょう。このページからドライバをダウンロードすることができます: http://www.oracle.com/technetwork/database/features/instant-client/index-097480.htmlシステムのアーキテクチャーに対応するLinux x86 か x86-64バージョンを選んでください。コマンド**uname -i**でこれをチェックすることができます。それは、x86-64のための「x86_64」あるいはx86のための「i386」のいずれかです。以下のパッケージをダウンロードするべきです：'Instant Client Package - Basic', 'Instant Client Package - SQL*Plus', and 'Instant Client Package - SDK'。それらをディスクに保存してください。ここで、rootユーザーとして、次のコマンドを使用して、パッケージをインストールすることができます:

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

この後、2つの環境変数をセットし、DBD::Oracleドライバーをコンパイルするべきです。再び、rootユーザとしてこれらのタスクを行なってください。以下にステップを概説します。コマンドが除去されたことで出力行が数行短くなっていることにお気づき下さい。

```
[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.
```

ここで、ORACLE_HOMEを提供するファイル　Kernel/Config.pmを編集するべきです。次のステップはこのセクションで記述されるように、ウェブ・インストーラを使用して、OTRSを構成することです。

1.3. Debian または Ubuntuシステム上でのOTRSのインストール

重要

OTRSはソースからインストールして下さい。Debain/Ubuntuが提供するOTRSパッケージを使用しないで下さい。

利用可能なパッケージを使用すれば、必要なPerlモジュールのインストールはより容易です:

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

1.4. Microsoft Windows システム上でのOTRSのインストール

1.4.1. Windowsインストーラの使用

OTRSをMicrosoft　Windowsシステムにインストールすることは非常に容易です。Win32のための最新のインストーラをhttp://www.otrs.com/try/からダウンロードし、ファイルをローカルのファイルシステムへ保存します。次にダブルクリックするだけでインストーラを実行し、システムをセットアップするためにインストレーションのわずかなステップを踏みます。インストレーション・ウィザードは、OTRSのためのすべてのコンポーネントをインストールしてくれます。次のステップはこのセクションで記述されるように、ウェブ・インストーラを使用して、OTRSを構成することです。

重要

OTRSのためのウィンドウズのインストーラは、OTRS(つまりApacheウェブサーバ、MySQLデータベース・サーバ、Perl(すべての必要とされるモジュールで)、ウィンドウズ用のcron) のための必要とされるすべてのコンポーネントを含んでいます。しかし、OTRSのセットアップは、Apacheの代わりにMicrosoft IISウェブサーバー上でもできます。また、例えば中枢のデータベース (MySQL、PostgreSQL、Microsoft SQL Serverあるいは Oralce) 上で展開したくないならば、バンドルされたMySQLをインストールさせない選択も可能です。

1.4.2. Windows上でのOracleデータベースのインストレーション

Windowsインストーラを使用して、OTRSをインストールすれば、それは典型的に必要なすべてのPerlモジュールを自動的に構成します。1つだけ例外があります:Oracleデータベース上でOTRSを

展開させる必要があれば、Oracleドライバを後からインストールする必要があります。利用可能なオラクルのためのオープン・ソース・ドライバがないので、これは必要です;私たちは、OTRSパッケージ中でそれを再分配することを認められません。Windows上で使用することができる2つの異なるperlがあります;バンドルされたStrawberry Perlを選ぶことができます。それはデフォルト・オプションで、概念実証のセットアップ用に推奨されます;あるいは、個別のダウンロードであるActive State perlを選びインストールすることができます。この後者のオプションは製品セットアップに推薦されます。

1.4.2.1. Installation of DBD::Oracle on ActiveState Perl

ActiveStatePerlをインストールしたら、既にPerlデータベース・ドライバDBD::Oracleが付属しており、これだけはOracle Cクライアントにリンクされるべきです。OTRSは32ビットのActiveStatePerlのみで稼働します。これはさらに、32ビットのOracleクライアントをダウンロードするべきであることも意味します。Oracleクライアントのウェブサイト*Instant Client Downloads for Microsoft Windows (32-bit)*そして、'Instant Client Package - Basic'を選択します。ソフトウェアをダウンロードするために無料のオラクル・ウェブ・アカウントにサインアップする必要があります。ドライブ(例えば`C:\instantclient_11_2`)のrootにzipファイルをダウンロードしてください。

ここで、パスにクライアント・ライブラリを追加し、環境変数'ORACLE_HOME'をセットしなければなりません。コントロールパネル> システムとセキュリティ> システムにナビゲーションして、「システム詳細設定」を選び、「システム変数」下の「新規」をクリックし、変数名として'ORACLE_HOME' を入力し、変数値として「C:\instantclient_11_2」を使ってください。「PATH」変数を選んでください、そして、パスにディレクトリ「C:\instantclient_11_2」を追加してください。ここで新しいcmd.exe セッションを開始し、OTRSディレクトリにあるotrs.CheckModules.plスクリプトを走らせればDBD::Oracle driverがうまくインストールされたことがわかるでしょう。この後に、他のアプリケーションが新しい環境変数を検知するために、システムを再開しなければなりません。

1.4.2.2. Strawberry Perl上での**DBD::Oracle** のインストレーション

DBD::Oracleモジュールをコンパイルするために、OTRSウェブ・インストーラのコンポーネントをスペースを含まないパスにインストールする必要があります。そのパスは`C:\Program Files\OTRS`ではだめで、`C:\OTRSComponents`の中か、あるいはそれに近いパスであるべきです。

Strawberry Perl では、オラクル・ドライバを手動でコンパイルするべきです。OTRS Windowsインストーラは32ビットのStrawberry Perlをバンドルします。これは、32ビットのオラクル・クライアントをダウンロードするべきであることを意味します。オラクル・クライアントのウェブサイト*Instant Client Downloads for Microsoft Windows (32-bit)*へナビゲートし、パッケージ'Instant Client Package - Basic'、'Instant Client Package - SQL*Plus'と'Instant Client Package - SDK'といったパッケージをダウンロードします。ソフトウェアをダウンロードするために無料のオラクル・ウェブ用アカウントにサインアップする必要があります。ダウンロードした全てのzipファイルを、ドライブのroot (例えば`C:\instantclient_11_2`) に抽出します。

ここで、クライアント・ライブラリをパスに追加し、環境変数'ORACLE_HOME'をセットするべきです。コントロール・パネル> システム> セキュリティ>にナビゲートし、「システム詳細設定」を選び、「システム変数」下の「新規」をクリックし、変数名として'ORACLE_HOME' を入力し、変数値として「C:\instantclient_11_2」を使ってください。「PATH」変数を選択して、パスにディレクトリ「C:\instantclient_11_2」を追加してください。この後、新しい環境変数を検知するために他のアプリケーションのためにシステムを再開するべきです。

ここで、DBD::Oracleモジュールをコンパイルしインストールすることができます。これについては、cmd.exeウィンドウを開けて、cpanクライアントを開始してください。**look DBD::Oracle**経由でDBD::Oracleダウンロードし抽出すします。その後**perl Makefile.PL**, **dmake** と **dmake install**をタイプします。サンプル出力は以下にあります。

```
C:\Users\max>cpan

cpan shell -- CPAN exploration and modules installation (v1.9800)
```

```
Enter 'h' for help.


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

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

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

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

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

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

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

cpan> quit
Lockfile removed.

C:\Users\max>
```

1.4.2.3. ウェブインターフェイスを使用した**OTRS**の構成

Webインストーラーを用いて、OTRSの基本的な構成の設定が行えますWebインストーラー.

2. ソース (Linux,Unix)からのインストレーション

2.1. ソースからのインストレーションの準備

ソースからOTRSをインストールしたければ、まず.tar.gz, .tar.bz2, or .zip file http://www.otrs.com/try/といったソース・アーカイブからダウンロードして下さい。

アーカイブをディレクトリー/optへ解凍して(例えば、**tar**を使用して)、ディレクトリーを otrs-x.x.x から otrs へ改名してください(下記のスクリプトを参照)。

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

スクリプト：*OTRS*をインストールするファースト・ステップ

OTRSは、root権限で実行されるべきではありません。今直ぐOTRSのための新規のユーザを追加すべきです。この新規ユーザのホームディレクトリーは/opt/otrsにあるべきです。ウェブサーバが新規の「otrs」ユーザ(それはほとんどのシステム上の場合である)と同じユーザー権利で作動していなければ、新規の「otrs」ユーザをウェブサーバ・ユーザのグループに追加する必要があります(下記のスクリプトを参照)。

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

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

スクリプト：新規 '*OTRS*' ユーザの追加とグループへの追加

次に、いくつかのサンプル構成ファイルをコピーしなければなりません。システムは後でコピーされたファイルを使用します。ファイルは、/opt/otrs/Kernel and /opt/otrs/Kernel/Config にあり、接尾辞.distがあります(下記のスクリプトを参照)。

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

スクリプト：いくつかのサンプルファイルのコピー

OTRSのインストレーションを準備する最後のステップは、ファイル用に適切なアクセス権を設定することです。「otrs」ユーザのホームディレクトリーの中で、スクリプト**otrs.SetPermissions.pl**(それはbinディレクトリーに位置する)を使用することができます。次のパラメーターを備えたスクリプトを実行することができます:

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

ウェブサーバがユーザ「otrs」と同じユーザー権限で作動している場合、適切なアクセス権を設定するコマンドは**otrs.SetPermissions.pl /opt/otrs --otrs-user=otrs --web-user=otrs**。SUSEシステムにおいては、ウェブサーバが「wwwrun」のユーザー権限で作動しています。Debianベースのシステムにおいては、これが「www-data」です。適切なアクセス権限を設定するためにコマンド**otrs.SetPermissions.pl /opt/otrs --otrs-user=otrs --web-user=wwwrun --otrs-group=nogroup --web-group=www**を使います。

2.2. Perlモジュールのインストレーション

テーブル3-1に述べられているように、OTRSはいくつかの追加のパール・モジュールを必要とします。ソースからOTRSをインストールすれば、これらのモジュールを手動でインストールする必要があります。これは、Linuxディストリビューションyast、apt-get)のパッケージ・マネージャーを、あるいはPerlシェルおよびCPANによって(このセクションに述べられていたように)やめることができます。ActiveState Perlを使用していれば、例えば、ウィンドウズにおいては、PPM(内蔵のPerl・パッケージ・マネージャー)を使用するかもしれませんが、私たちはできればパッケージ・マネージャーを使用することを推奨します。

otrs.CheckModules.plでどのモジュールをインストールする必要があるか確認することができます。このスクリプトは「otrs」ユーザのホームディレクトリーに、binディレクトリーに位置します(下記のスクリプトを参照)。いくつかのモジュールはオプションであることに注意してください。

```
linux:~# cd /opt/otrs/bin/
linux:/opt/otrs/bin# ./otrs.CheckModules.pl
```

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

  ...
```

スクリプト：必要なモジュールのチェック

Linuxディストリビューションのパッケージ管理システムから不足しているモジュールをインストールするために努力するべきです。そうすることによって、新バージョンが利用可能な場合、あるいはセキュリティ問題が見つかった時、パッケージは自動的に更新されます。追加のパッケージをインストールする方法についての配布ドキュメンテーションを参照してください。（正しいバージョンの）モジュールがパッケージ・リポジトリから利用可能でない場合、CPAN、包括的なPerl・アーカイブ・ネットワークからもインストールすることができます。

上記のCPAN経由でモジュールのうちの1つをインストールするために、コマンド**perl -e shell -MCPAN**を実行しなければなりません。Perlシェルは会話型の中で開始され、CPANモジュールがロードされます。CPANが既に構成されている場合、モジュールの名前に続いて、コマンド**install**でモジュールをインストールすることができます。CPANは、他のPerl・モジュールにモジュールの依存性に気を配り、他のモジュールが必要かどうかを知らせます。

コマンド**perl -cw bin/cgi-bin/index.plperl -cw bin/cgi-bin/customer.pl** を実行し、ディレクトリ/opt/otrsに変更した後で、**perl -cw bin/otrs.PostMaster.pl**を実行します。 両方のコマンドの出力が"シンタックス OK"である場合、Perlは適切にセット・アップされます(下記のスクリプトを参照)。

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

スクリプト：シンタックス・チェック

2.3. Apacheウェブサーバの構成

第一に、Apache2ウェブサーバおよびmod_perlをインストールするべきです;典型的にはシステム・パッケージ・マネージャーからこれをします。以下の、最も人気のあるLinuxディストリビューション上でApacheを設定するために必要なコマンドをご覧ください。

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

# suse:
linux:# zypper install apache2-mod_perl

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

短いURL、エイリアスおよびScriptAliasエントリー経由でOTRSのウェブ・インターフェースにアクセスすることは必要です。ほとんどのApacheインストレーションはconf.dディレクトリを含みます。Linuxシステムにおいては、通常/etc/apacheあるいは/etc/apache2の下にこのディレクトリーを見つけることができます。rootとしてログインして、conf.dディレクトリーを変更し

て、アパッチ構成ディレクトリー中のotrs.confと呼ばれるファイルに/opt/otrs/scripts/apache2-httpd.include.confの中の適切なテンプレートをコピーしてください。

新しい構成設定をロードするためにウェブサーバを再起動してください。ほとんどのシステムにおいては、コマンド**/etc/init.d/apache2 restart** でウェブサーバを開始/再起動することができます(下記のスクリプトを参照)。

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

スクリプト：ウェブサーバの再起動

ここで、ウェブサーバはOTRS用に構成されるべきです。

2.4. データベースの構成

2.4.1. OTRSデータベースのマニュアル・インストール

データベースを構成する推奨方法は、Web　Installerを走らせることです。何らかの理由でこれを使用することができなければ、本章に述べられているように、データベースを手動で構成することもできます。OTRSデータベースをセットアップするためのウェブ・インストーラーを使用することができない場合、それを手動でセット・アップしなければなりません。データベースを作成し構成するSQLステートメントを備えたスクリプトは「otrs」ユーザのホームディレクトリーに、scripts/databaseに位置します(下記のスクリプトを参照)。

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

スクリプト：データベースを作成し構成するために必要なファイル

異なるデータベースバックエンド用にデータベースをセットアップするために、.sqlファイルは特殊な順番で処理されなければなりません。

OTRSデータベースを手動でステップバイステプで作成します

1. DBの作成:データベース・クライアントあるいはデータベース・インターフェースで、OTRSのために使用したいデータベースを作成してください。

2. テーブルの作成:　otrs-schema.DatabaseType.sqlファイル(例えばotrs-schema.oracle.sql, otrs-schema.postgresql.sql)で、OTRSデータベースにテーブルを作成することができます。

3. 初回のシステムデータの挿入:　OTRSは、適切に稼働するためにある初回のシステムデータ (例えば異なるチケット状態、チケットとその通知タイプ)を必要とします。使用しているデータベースのタイプによって、次のファイルのうちの1つを使用する必要があります:otrs-initial_insert.mysql.sql,　otrs-initial_insert.oracle.sql,　otrs-initial_insert.postgresql.sql あるいは otrs-initial_insert.mssql.sql 。

4. テーブル間の参照の作成:　最後のステップは、OTRSデータベースに異なるテーブル間の参照を作成することです。これらを作成するためにotrs-schema-post.DatabaseType.sql　ファ

イルを使用します。(例えば. otrs-schema-oracle.post.sql, otrs-schema-post.postgresql.sql)。

データベース・セットアップを終了後、OTRSデータベース用の適切なアクセス権をチェックし設定するべきです。1人のユーザにアクセスを許可することで十分です。使用しているデータベース・サーバによって、アクセス権の設定は異なります。しかし、それは、データベース・クライアントあるいはグラフィカルなデータベースのフロントエンドのいずれかで可能であるべきです。

データベースとアクセス権が適切に構成される場合、どのデータベース・バックエンドを使用したいか、また、チケット・システムがどのようにデータベースに接続することができるかOTRSに伝えなければなりません。「otrs」ユーザのホームディレクトリにあるファイルKernel/Config.pmを開き、必要に応じてスクリプトの中で下に示されるパラメーターを変更してください。

```
# DatabaseHost
# (データベースホスト.)
$Self->{'DatabaseHost'} = 'localhost';

# Database
# (データベース名.)
$Self->{Database} = 'otrs';

# DatabaseUser
# (データベースユーザ.)
$Self->{DatabaseUser} = 'otrs';

# DatabasePw
# (データベースユーザのパスワード.)
$Self->{DatabasePw} = 'some-pass';
```

スクリプト：カスタマイズ用パラメタ

2.5. OTRS用cronジョブのセットアップ

OTRSは、適切に稼働するいくつかのcronジョブを必要とします。cronジョブは、OTRSモジュール用に指定されたのと同じユーザー権利で実行されるべきです。それは、cronジョブが「otrs」ユーザのcrontabファイルに挿入されなければならないことを意味します。

cronジョブを備えたスクリプトはすべて「otrs」ユーザのホームディレクトリーに、var/cronにあります(下記のスクリプトを参照)。

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

スクリプト：*cron*ジョブを作成する必要のあるファイル

これらのスクリプトには「 . dist」の接尾辞があります。接尾辞を削除して、ファイルにそれらをコピーするべきです。bashを使用すれば、以下のスクリプトにリストされたコマンドを使用したいと思うかもしれません。

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

```
fetchmail.dist          postmaster.dist             unlock.dist
generic_agent           postmaster_mailbox
generic_agent-database  postmaster_mailbox.dist
linux:/opt/otrs/var/cron#
```

スクリプト：*cron*ジョブを作成する必要のある全てのファイルをコピーして改名する

テーブル 3-2 は異なるcronジョブを記述しています。

表**2.1** いくつかの**cron**ジョブ・スクリプトの記述

Script	**Function**
aaa_base	「otrs」ユーザのcrontab用基本を設定します。
cache	ディスクから失効したキャッシュを削除します。CCS及びJavaScript用ローダー・キャッシュをクリアします
fetchmail	新しいメールがチケット・システムの中へのfetchmailで取って来られる場合のみ使用されます。
generic_agent	データベースに格納されず、自身のconfigファイルにあるGenericAgentのジョブを実行します。
generic_agent-database	データベースに格納されるGenericAgentのジョブを実行します。
pending_jobs	保留チケット用システムをチェックし、それらを完了します。あるいは必要ならばリマインダを送信します。
postmaster	チケット・システムのメッセージ・キューをチェックし、キューにまだあるメッセージを配布します。
postmaster_mailbox	「PostMasterメールアカウント」用セクションの管理者エリアで指定されたPOP3アカウントからメールを取ってきます。
rebuild_ticket_index	キュー・ビューの速度を改善するチケット・インデックスを再構築します。
session	古くもう必要のないセクションIDを削除します。
unlock	システムのチケットをアンロック

セットアップに、すべてのcronジョブ、「otrs」ユーザのホームディレクトリーにあるスクリプトbin/Cron.shは使用することができます。このスクリプトを実行する場合、cronジョブをインストールするか、削除するか、再インストールするかどうかを明示するためにパラメーターを必要とします。次のパラメーターを使用することができます:

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

cronジョブは「otrs」ユーザのcrontabファイルにインストールする必要があるので、「otrs」としてログインする必要があります。rootとしてログインする場合は、**su　otrs**で「otrs」に変わることができます。cronジョブをインストールするために下記スクリプト中で指定されたコマンドを実行してください。

警告

「otrs」ユーザの他のcrontabエントリーがCron.shスクリプトによって上書きされるか削除されるだろうことに注意してください。必要な他のcrontabエントリーを保持するためにCron.shスクリプトを変更してください。

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

スクリプト：*cron*ジョブのインストール

rootとして実行可能なコマンド**crontab -l -u otrs**は、「otrs」ユーザのcrontabファイルを示します。また、全てのエントリーが正確に置かれるかどうかをチェックすることができます(下記のスクリプトを参照)。

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

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

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

# --
# cron/generic_agent - otrs.GenericAgent.pl cron of the OTRS
# Copyright (C) 2001-2014 OTRS AG, http://otrs.com/
# --
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# start generic agent every 20 minutes
*/20 * * * * $HOME/bin/GenericAgent.pl >> /dev/null
# example to execute GenericAgent.pl on 23:00 with
# Kernel::Config::GenericAgentMove job file
#0 23 * * * $HOME/bin/otrs.GenericAgent.pl -c "Kernel::Config::GenericAgentMove" >> /dev/
null
# --
# cron/generic_agent - GenericAgent.pl cron of the OTRS
# Copyright (C) 2001-2014 OTRS AG, http://otrs.com/
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# start generic agent every 10 minutes
*/10 * * * * $HOME/bin/otrs.GenericAgent.pl -c db >> /dev/null
# --
# cron/pending_jobs - pending_jobs cron of the OTRS
# Copyright (C) 2001-2014 OTRS AG, http://otrs.com/
```

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

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

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

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

スクリプト：*Cron*タブ・ファイル

2.6. OTRSスケジューラ・サービス

OTRSには、非同期タスクを行なうために利用されるスケジューラ・サービスが付属しています。

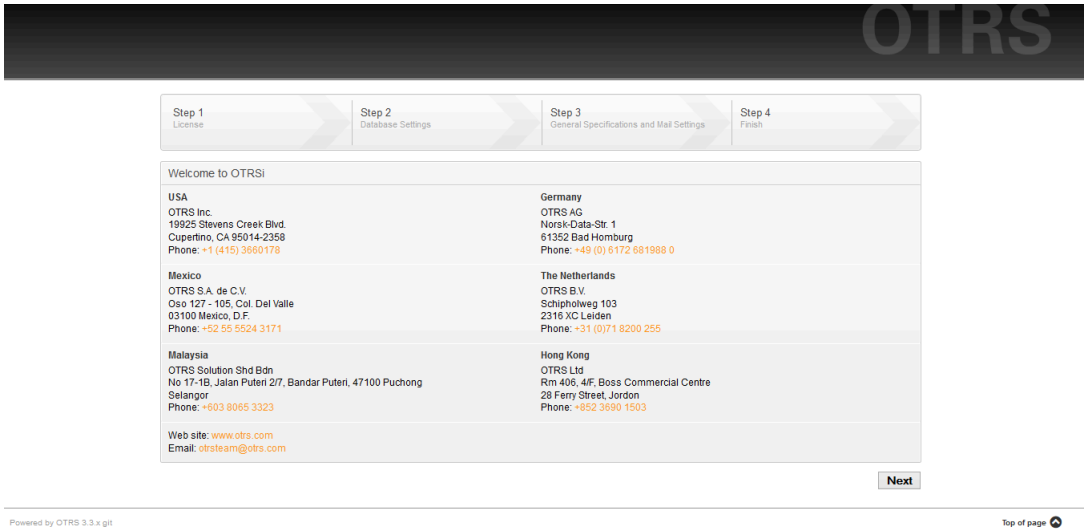
OTRS RPMはスケジューラ・サービスを自動的にセット・アップします。ソースからインストールすれば、/etc/init.dにscripts/otrs-scheduler-linuxファイルをコピーしそれに適切なアクセス許可を与えることにより、サービスをインストールすることができます。これで、システム開始時には必ずスケジューラ・サービスが開始するようになります。

3. ウェブ・インストーラの使用

OTRSデータベースをセット・アップおよび構成するためにOTRSソフトウェアをインストールした後、OTRSウェブ・インストーラを使用することができます。ウェブ・インストーラはブラウザから訪れることができるウェブ・ページです。ウェブ・インストーラ用のURLはhttp://localhost/otrs/installer.plです。

ウェブ・インストーラが開始したらシステムをセットアップする次のステップにしたがって下さい。

1. OTRSオフィスに関する情報をチェックして、次に進むをクリックします (下図参照) 。



図：ウェルカム画面

2 GNU Affero General Public Licenseを読み (下図参照) 、ページ下部にある対応するボタンをクリックしてそれを受諾します。

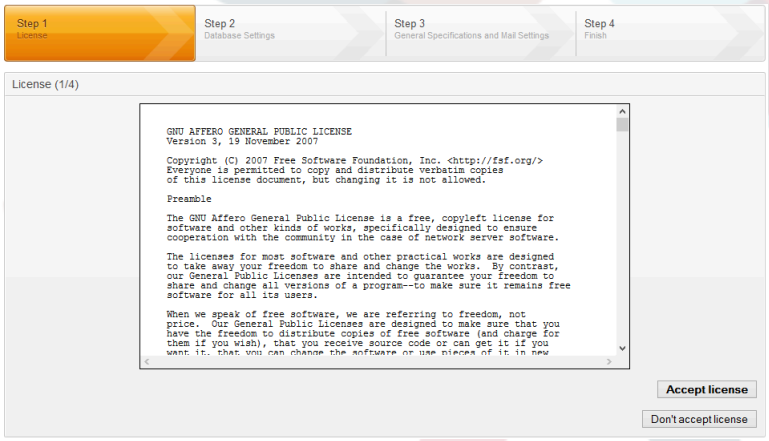
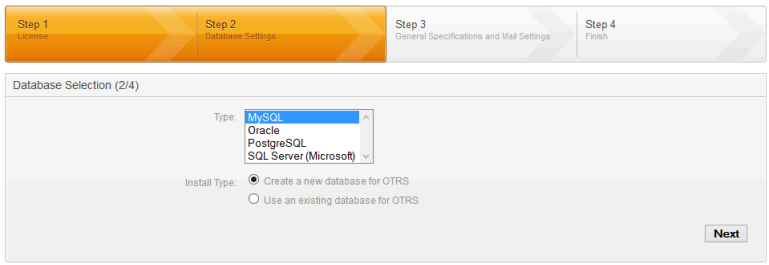


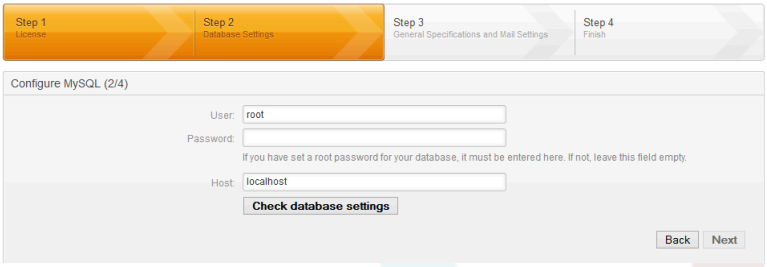
図: *GNU Affero General Public* ライセンス

3 OTRSで使用したいデータベースを選択します。データベースとして、MySQL、PostgreSQL あるいは Microsoft SQL Serverを選択する場合、また、データベースを作成するためにウェブ・インストーラーが必要ならば、あるいはデータベース管理者が、使用したい空のデータベースを既に作成していれば、ここで選択することもできます。その後、次ボタンをクリックしてください(下記の図を参照)。



図：データベース選択

4. 選んだデータベースによって、および、データベースを作成するためにウェブ・インストーラを必要とするか、または、前ステップの既存のものを使用するか、によって、この画面は少し異なるかもしれません。この画面でデータベース用に認証情報を入力してください。

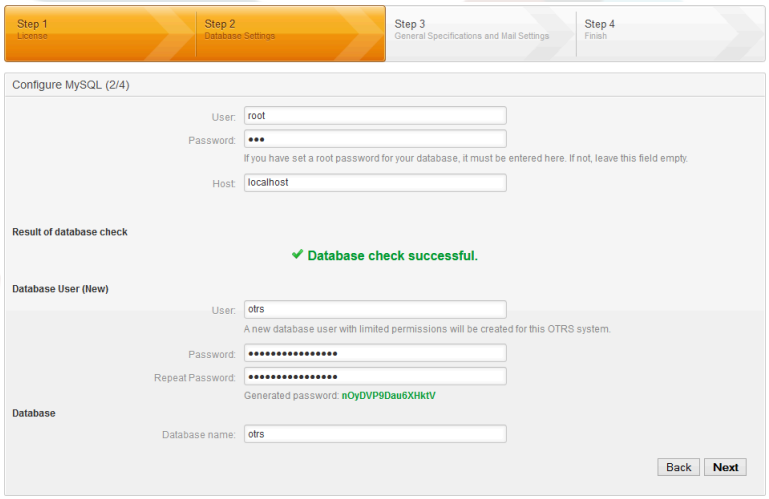


図：データベース認証情報

5. 新規データベースユーザを作成し、データベース名を選択し、「次」をクリックして下さい (下図参照) 。

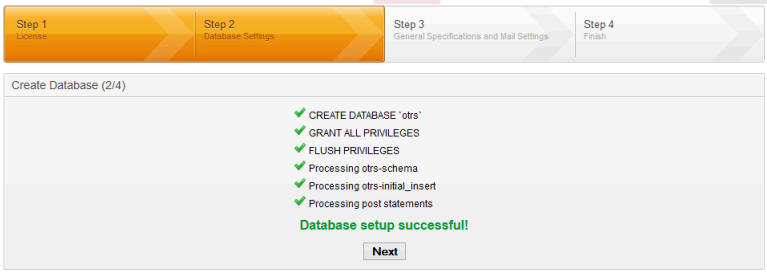
警告

OTRSは、強いパスワードを生成します。望めば自分のパスワードを入力することも可能です。パスワードは構成ファイルKernel/Config.pmに書かれるのでこのパスワードを思い出す必要はありません。



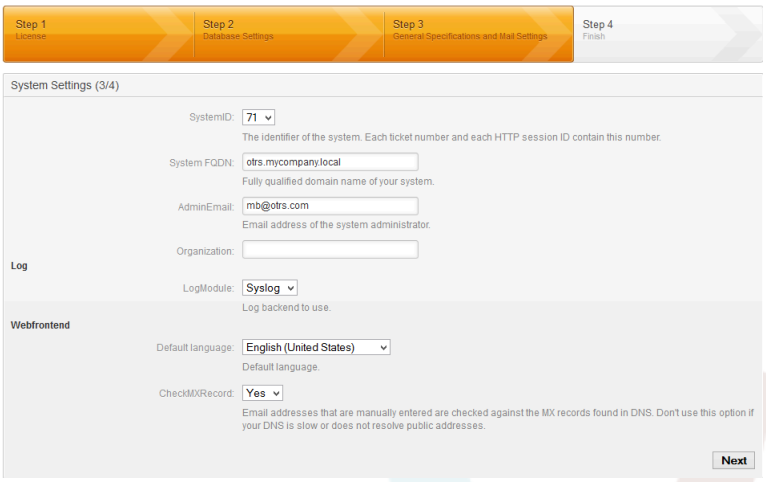
図：データベース設定

6. 必要ならデータベースは作成され、この図で示されるように追加されます。次のスクリーンに行くために「次へ」をクリックしてください。



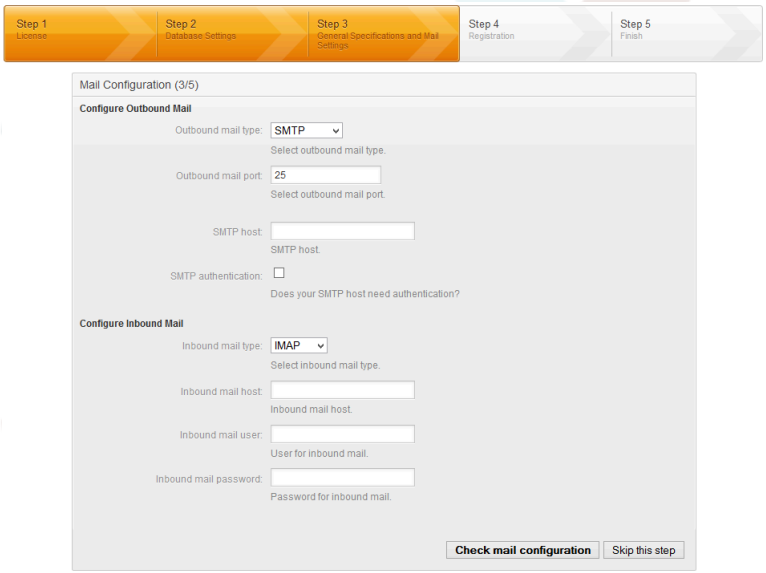
図：データベース設定の成功

7. 必要なすべてのシステム設定を提供し、「次に」をクリックしてください(下図参照)。



図：システム設定

8. 希望するならば、受信、発信メールを構成する必要なデータを提供し、あるいは、画面の下部を右クリックを押すことでこのステップを割愛します。(下図参照)。

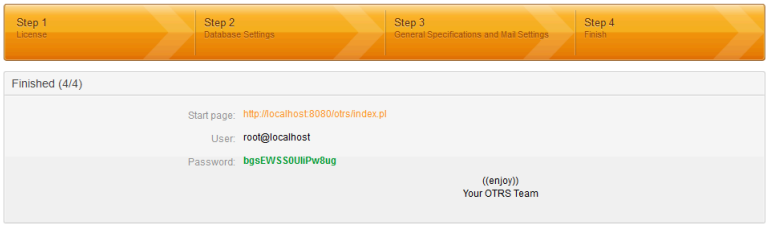


図：メール構成

9. おめでとうございます! これで、OTRSのインストレーションは終了です。そして、システムで稼働できるに違いありません(下記の図を参照)。OTRSのウェブ・インターフェースにログインするために、ブラウザからアドレスhttp://localhost/otrs/index.plを使用します。ユーザー名 'root@localhost'　および生成されたパスワードを使用して、OTRS管理者としてログインしてください。その後、要求を満たすシステムを構成することができます。

警告

'root@localhost' アカウント用に生成したパスワードを書き留めておいて下さい。



図：ウェブ・インストラー最終画面

4. OTRSフレームワークのアップグレード

これらは、OTRSをバージョン*3.2*から*3.3*へアップグレードする方々のための指示書です。RPMとソースコード（tarball)アップグレードの両方に適用されます。

注記

OTRS 3.3は少なくともperlバージョン5.10.0を必要とすることに注意してください。このバージョンを稼働するサーバアップグレードする計画する前にご確認ください。コマンドラインからコマンド**perl -v**でバージョンをチェックすることができます。perl 5.8を使用し、まだベンダーにサポートされている既知のLinuxディストリビューションは、ed Hat Enterprise Linux (RHEL) 5です。また、そのコミュニティーは派生的なCentOS 5をサポートしました。これらのプラットフォームのうちのどの上にもいて、OTRS 3.3にアップグレードすることを計画すれば、さらに、RHEL 6あるいはCentOS 6のような支援されたperlを備えたバージョンへのオペレーティング・システムを移動させて計画するべきです。

OTRSの旧バージョンをご利用の場合は、まず3.1までアップグレード順に従わなければなりません。 (1.1->1.2->1.3-> 2.0-> 2.1-> 2.2-> 2.3-> 2.4-> 3.0-> 3.1-> 3.2-> 3.3 ...)!データベース変更およびperlスクリプトのアップグレードを含めて、すべてのバージョンへの十分なアップグレードを実行する必要があります。

OTRS 2.2以前からアップグレードする場合は、余計な手間を踏む必要があります：AA。を読んでください。 http://bugs.otrs.org/show_bug.cgi?id=6798を読んで下さい。

単一の小さなバージョン内では、アップグレードしたければ、パッチ・レベル・リリースをスキップすることができます。例えば、OTRS 3.3.1からバージョン3.3.4に直接アップグレードすることができます。そのような"パッチ・レベル・アップグレード"を行う必要があれば、ステップ9および13-16をスキップするべきです。

OTRS 3.3.6以前からのパッチ・レベル・アップグレードを行なうならば、 スケジューラ・プロセスをコントロールする新規cronジョブを活性化する必要があることに注意してください。ファイル `/opt/otrs/var/cron/scheduler_watchdog.dist` を`/opt/otrs/var/cron/scheduler_watchdog` にコピーして`/opt/otrs/bin/Cron.sh start` でcrontabを更新して下さい(rootではなく"otrs" ユーザとして)。

1. 全ての関連サービスを停止

   例えば（使用中のサービスに依存しますが）：

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

2. 以下の全てをバックアップ $OTRS_HOME (default: OTRS_HOME=/opt/otrs):

   - `Kernel/Config.pm`
   - `Kernel/Config/GenericAgent.pm`
   - `Kernel/Config/Files/ZZZAuto.pm`
   - `var/*`
   - `as well as the database`

3. 全てバックアップしたことを確認して下さい。

4. 新規システム (オプション) のセットアップ

   可能ならばまずテスト用にこれを別マシンでインストールしてみて下さい。

5. 新規リリースをインストール (tarまたはPRM)。

   - talballで：

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

     旧構成ファイルをリストア

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

     TicketCounter.logをリストア

     OTRSにてチケット番号を正確に継続させるために、$OTRS_HOME/var/log/ (default: OTRS_HOME=/opt/otrs)にTicketCounter.logをリストアしてください。インクリメントのチケット番号を使用しているならば、特にこれは重要です。

     記事データのリストア

     OTRSを構成し、ファイルシステムに記事データを格納するならば、$OTRS_HOME/var/ (default: OTRS_HOME=/opt/otrs)にarticleフォルダーをリストアしなければなりません。

   - RPMで：

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

     この場合、RPMアップグレードは自動で旧構成ファイルをリストアします。

6. 自身のテーマ

   ノート: OTRSテーマの3.2と3.3は互換性がありません。だから旧テーマを使用しないで下さい。

   テーマは $OTRS_HOME/Kernel/Output/HTML/*/*.dtl (default: OTRS_HOME=/opt/otrs) 下にあります。

   dtl-FilesはOTRS 3.3上ではutf-8でなければいけません。

7. ファイル権限を設定

   talballを使用する場合は、システム設定に必要な権限で

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

   を実行します。

8. 必要な perlモジュールをチェックします。

システムに必要とされるperlモジュールがすべてインストールされ不足しているおそれのあるモジュールをインストールすることを確認してください。

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

9. スキーマ・アップデート：

ノート: MySQL UPGRADINGプロセスで作成された新しいテーブルは、MySQLサーバーで設定されたデフォルト・テーブル・ストレージ・エンジンで作成されます。MySQL 5.5では、新しいデフォルト・タイプはInnoDBです。既存のテーブル(例えば「ユーザ」)にテーブル・ストレージ・エンジンがある場合（例えばMyISAM）外部キー制約を作成するとエラーが表示されます。

2つのオプションがあります: (1) 新規テーブルが既存テーブルと同じエンジンを持つように、MySQLのデフォルトのストレージ・エンジンをMyISAMに変更できます。あるいは (2) ストレージ・エンジンとしてInnoDBを使用する既存テーブルを変更します。

otrs.CheckDB.plスクリプトは、ストレージ・エンジンに関するどんな問題も報告するので、可能な問題をチェックするためにそれを実行してください。

```
shell> cd /opt/otrs/

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

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

# PostgreSQL, older versions:
shell> cat scripts/DBUpdate-to-3.3.postgresql_before_8_2.sql | psql --set
 ON_ERROR_STOP=on --single-transaction otrs otrs
```

ノート:PostgreSQL 8.1以前を使用する場合は、これらのより古いバージョン用の新しいレガシー・ドライバーを活性化する必要があります。そのためには、新規の１行を

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

のようなKernel/Config.pmに追加します。

統合スクリプト（「root」としてではなくユーザ「otrs」として）を走らせます。

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

このスクリプトが適切に動かない場合はアップグレードプロセスを継続しないで下さい。そうでなければデータを失いかねません。

10構成をリフレッシュしてキャッシュを削除します。（「root」としてではなくユーザ「otrs」として）

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

11サービスの再起動

例えば（ご使用中のサービスによりますが）

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

今、システムにログインできます。

12インストール済のパッケージをチェック

パッケージ・マネージャでは全てのパッケージが正しくインストールされたとしてマークされているか、あるいはなんらかの再インストレーションあるいはパッケージ・アップグレードでさえ必要であるかをチェックします。

以下のパッケージはアップグレード・プロセス後。自動的にアンインストールされます (先にインストール済ならば)

- OTRSPostMasterFilterExtensions
- OTRSFreeTextFromCustomerUser
- OTRSExternalTicketNumberRecognition
- OTRSDashboardQueueOverview
- OTRSImportantArticles
- OTRSImportantArticlesITSM
- OTRSDashboardTicketCalendar
- OTRSMultiServiceSelect
- OTRSMultiQueueSelect
- OTRSDynamicFieldMultiLevelSelection
- OTRSDynamicFieldMultiLevelSelection
- OTRSTicketAclEditor
- OTRSCustomerProcessSelection
- OTRSACLExtensions
- OTRSGenericStandardTemplates
- OTRSExtendedDynamicDateFieldSearch
- OTRSDashboardTicketOverviewFilters

13OTRSFreeTextFromCustomerUserの構成設定をチェックします。

ノート: これは以前にOTRSFreeTextFromCustomerUser パッケージを使用した場合にだけ適用されます。

このモジュールを以前に使用したならば、それを再構成する必要がああります。それが今OTRSフレームワークの一部であるように、モジュールはアップグレードするスクリプトによって自動的にアンインストールされます。

それを使用し続けたい場合は、この機能を活性化するために、設定 "Ticket::EventModulePost###930-DynamicFieldFromCustomerUser" を有効にして下さい。そして設定 "DynamicFieldFromCustomerUser::Mapping"でマッピングを構成して下さい。

14ACLを新規ALCエディタ（オプション）へインポートします。

OTRS　3.3では、管理インターフェースにACL用グラフィカルなエディタがあります。エディタに既存のACLを利用可能にするために、それらを（例えばConfig.pmあるいは追加ファイル）をbin/otrs.ImportACLsFromConfig.pl を使用することによりエディタにインポートする必要があります。インポート手続きをうまく終了させた後では、必ずConfig.pm(または他のファイル)からいかなるACLも削除して下さい。さらに、システムにインポートしたACLを再度展開させるためにACL管理フロント・エンドの中で展開ボタンを使用する必要もあります。

15ウェブサーバ構成のアップデート

ノート: Apacheウェブサーバを使用し、OTRSインストール・ディレクトリ(例えばApache 構成ディレクトリからのシンボリックリンクで)から設定ファイルを直接使用しない場合のみ、これは当てはまります。

いくつかの変更をした時は、OTRS用Apache設定ファイルをアップデートして下さい。(`scripts/apache2-httpd.include.conf`を御覧ください)。

16活性化しているcronジョブのアップデート

$OTRS_HOME/var/cron/*.distにいくつかのOTRSデフォルトのcronジョブがあります。それらは ".dist" のファイル拡張子なしでそれらをコピーすることで活性化することができますこれを実行する際は、新規cronjobsおよびcronjobsの最新バージョンの入手をご確認下さい。

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

コピーされたファイルをチェックして、カスタマイゼーション済ならすべて再適用してください。

システム上のこれらのcronジョブを予定するために、スクリプトCron.shを使用することができます。必ずOTRSシステム・ユーザとしてそれを実行してください!

```
shell&gt; /opt/otrs/bin/Cron.sh start
```

17お疲れ様でした！

5. Windowsインストーラのアップグレード

OTRSをWindowsにインストールしたならば、新しいWindowsインストーラのダウンロードしそれの実行することでシームレスにアップグレードすることが可能です。インストーラは自動的にOTRSのインストレーションを検知し、その上でアップグレードを行ないます。インストーラはOTRSサービスを停止しインストレーションが完了後、それらを開始します。そのためユーザにダウンタイムが発生するでしょう。必ずいつかアップグレードの計画を立ててください。アップグレード手続きを始める前に必ず常にサーバのバックアップを保持していることを確認ください!

5.1. Windows上でパッチ・レベルのアップグレードを実施します。

Windows にパッチ・レベル・アップグレード(つまりOTRS 3.3.1からOTRS 3.3.2まで)をインストールする必要があれば、新規インストーラーをダウンロードし実行するだけです。もちろん、アップグレードを開始する前に必ずバックアップを保持していることを確認して下さい!

5.2. Windows上でのマイナー・レベルのアップグレードの実施

マイナー・アップグレード(OTRS 3.2からOTRS 3.3まで)は、データベース・アップグレードのステップも含むので多少時間がかかります。インストーラーはこれを検知し、自動的にアップグレードを行います。インストレーション後、さらに「正常系(regular)」upgrading sectionのステップ12、13および14を行なうべきです。

OTRSシステムにインストールした機能アドオンのようないかなる追加のパッケージでも持っていれば、同様にこれらをアップグレードする必要があることに注意してください。アップグレードを始める前にこれらの入手を確かめてください。アップグレード完了後、パッケージ・インストーラー・ウェブGUIによってこれらのパッケージをアップグレードできます。フレームワークアップグレードが完了していた後、サーベイ、FAQおよびITSMのような「標準」リポジトリからのパッケージは、パッケージ・マネージャーからアップグレードすることができます。

6. 追加のアプリケーション

OTRSフレームワークの機能性を拡張するために追加出願パッケージをインストールすることができます。これは、管理者ページからのパッケージ・マネージャーによって行うことができます。それはオンライン・リポジトリから適用をダウンロードし、パッケージ依存性を管理します。また、ローカル・ファイルからパッケージをインストールすることも可能です。

6.1. FAQ (よくある質問)

FAQは知識ベース・コンポーネントです。それは、FAQ記事の閲覧と編集をサポートします。記事は閲覧され、担当者、顧客ユーザあるいは匿名ユーザーを制限することができます。これらもグループへ組み立てて、異なる言語で読むことができます。

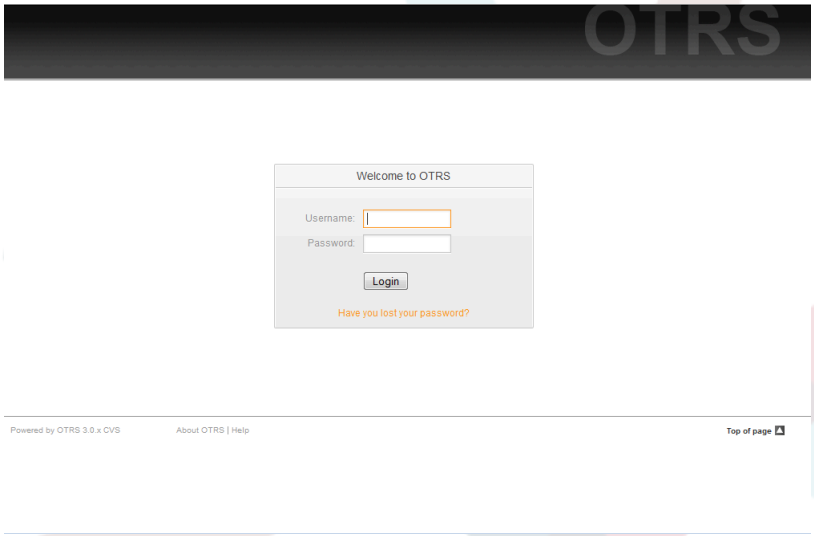
第3章 ファースト・ステップ

この章の目的は、OTRSの簡潔な概要を提供することです。

1. 担当者ウェブインターフェイス

担当者は、担当者ウェブインターフェイスで顧客のリクエストに答えることができます。顧客のためあるいは他の担当者のために新規チケットを作成し、顧客との通話に関することをチケットに書き、FAQを記入し、顧客データを編集します。

URL *http://www.example.com*経由であなたのOTRSホストへ到達すると仮定すれば、ウェブブラウザ（下図参照）でアドレスhttp://www.example.com/otrs/index.plを使用することでOTRSログイン画面へ到達します。

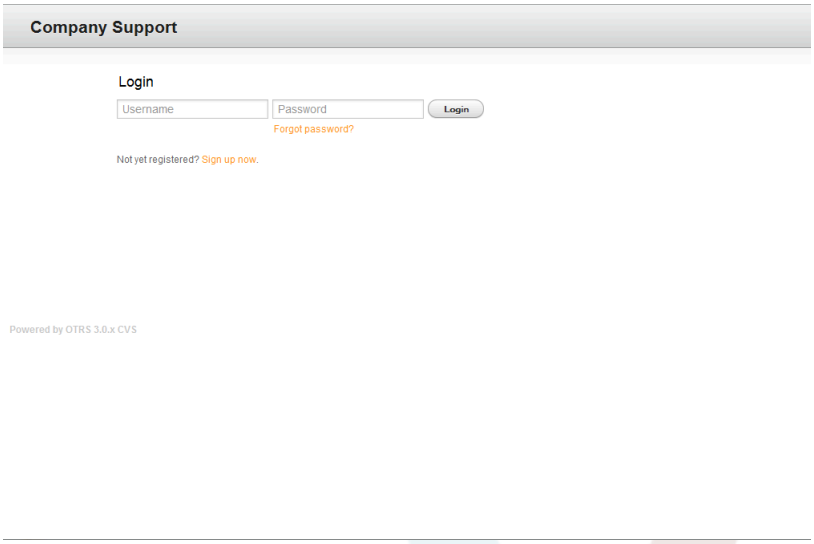


図：担当者インターフェイスのログイン画面

2. 顧客ウェブインターフェイス

顧客は、OTRSの別々のウェブインターフェイスを持ち、それを通じて、新規アカウントを作成し、アカウント設定を変更し、チケットを作成、編集し、作成したチケットに関する概要を取得することができます。

上のサンプルを続け、顧客ログイン画面はウェブブラウザでURLhttp://www.example.com/otrs/customer.plを使用することで到達します。（下図参照）

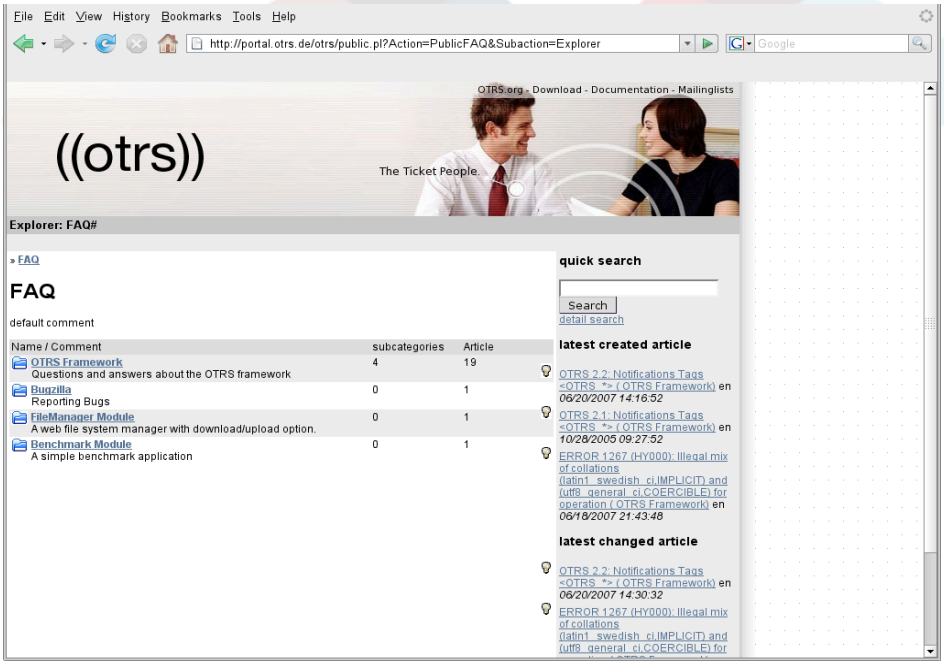


図：顧客ログイン画面

3. 公開ウェブインターフェイス

担当者と顧客のためのウェブインターフェイスに加えて、OTRSにはさらにFAQモジュールによって利用可能な公開ウェブ・インターフェースがあります。このモジュールを別々にインストールしている必要があります。それはFAQシステムに公衆アクセスを提供し、特別の認可のないFAQエントリーをビジターに捜させます。

私達のサンプルでは, 公開ウェブインターフェイスは次のいずれかのURL : http://www.example.com/otrs/faq.pl , http://www.example.com/otrs/public.plに到達できます。



図：パブリック・ウェブインターフェイス

4. はじめてのログイン

セクションAgent web interface に記載のようにログイン画面へアクセスします。ユーザ名とパスワードを入力します。システムはインストールされたばかりでユーザはまだ作成されていませんの

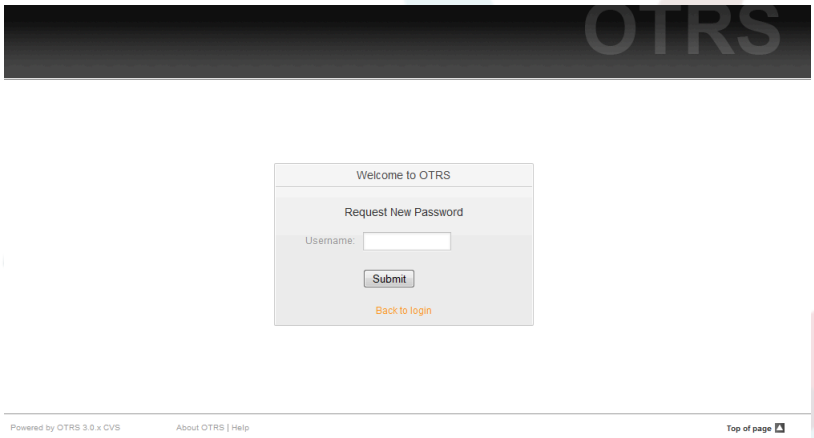
で、まず、ユーザ名として'root@localhost'　をパスワードとして'root'を使用してOTRS管理者としてログインします。

警告

このアカウントデータは
　どの新規にインストールされたOTRSシステムでも有効です。OTRS管理者パスワードはできるだけ早く変更すべきです。OTRS管理者用画面preferencesから実行できます。

OTRS管理者としてログインしたくなければ、あなたの通常の担当者アカウント用のユーザ名とパスワードを入力してもいいです

パスワードをお忘れの場合、新規パスワードをシステムに要求することができます。単にログインボタンの下のリンクを押すだけです。OTRSアカウント用の入力フィールドに登録されたEメールアドレスを入力します。送信ボタンを押します (下図参照) 。



図：新規パスワードの要求

5. ウェブインターフェイス - 概要

システムへログイン成功時、ダッシュボード・ページが表示されます(下記の図を参照)。ロックされたチケットを表示し、キュー、ステータスおよびエスカレーション・ビューへのメニューから直接アクセスを許可し、さらに、新規の電話および電子メール・チケットの生成に対するオプションも保持します。さらに様々な基準を使用して、チケットのクイックサマリも表示します。

図：初めてのログイン成功後の画面

わかりやすさを改善するために、一般的ウェブ・インターフェースは異なるエリアへ分離されています。各ページのトップの列は以下のようないくつかの一般情報を示します。現在のユーザー名、ログアウト・ボタン、ダイレクトアクセスを備えたロックされたチケットの数のリストアイコン、新しい電話/電子メール・チケットなどを作成するリンク。さらにキュー、ステータスおよびエスカレーション・ビューに行くアイコンがあります。

アイコンの下の行はナビゲーション・バーです。システムのモジュールや異なるエリアへナビゲートしてくれるメニューを表示し、いくつかの全体的アクションを実行させてくれます。ダッシュボード・ボタンをクリックすることでダッシュボード行けます。チケット・ボタンをクリックすれば、チケット・ビューを変更したり、新しいチケット(電話/Eメール)を作成したり、あるいは特定のチケットを探索したりするオプションを備えたサブメニューを使えます。統計ボタンは、新しいものを作成し、既存のものを輸入して、登記済みの統計の概観からあなたが選ぶことを可能にするメニューを表示します。顧客ボタンはあなたを顧客管理スクリーンへ導きます。管理者ボタンのクリックによって、管理者モジュールすべてにアクセスすることができます、それは新しい代理人、キューなどを作ることを可能にします。さらにチケット探索をするサーチ・ボタンもあります。

関連するアプリケーションがインストールされている場合、例えばFAQ、サーベイ、これらのアプリケーションを起動するボタンも表示されます。

ナビゲーション・バー下の赤いバーは様々なシステム・メッセージを表示します。OTRS管理者としてログインすると、このシステム・アカウントを使用して作業しない旨の警告メッセージを受けます。

現在表示しているセクションのタイトルの下に、いくつかのサブセクションがあります。別々のボックスに各 1 つづつ、作業中の画面に関する関連情報を含んでいます。

これらのボックスは各スクリーンの主要部分を含んでいます。それらは常に1つあるいはいくつかのカラムで表示されます。ボックスはそれぞれ、現在のスクリーンに関する関連情報 (例えば指示、アドバイス、概観など) を格納することができます。さらに、各画面に関連するアクションを実行するために必要なフォームあるいはツールも表示されます。例えば、レコードの追加、更新、削除、ログのチェック、構成設定の変更等のアクションです。

最後に、ページの最下部に、サイト・フッタが表示されます(下図参照)。それは、直接OTRS公式ウェブサイトにアクセスするかあるいはページのトップへ行くリンクを含んでいます。

通常は、アイコン列、ナビゲーション・バーおよびフッタはウェブ・インターフェース上のすべての画面の中で同じに存在します。

図：フッター

6. ダッシュボード

ダッシュボードはシステムのメインページです。ここでチケット・アクティビティに関連する他のスタッフやチケットに関する概要を得ることができます。それは担当者にとっての日常業務のための出発点であると言えます。それはデフォルトでチケットの即時サマリーを表示します。サマリーには他の情報の中で保留、エスカレーション、新規および対応中の情報があります。

ダッシュボードの最も重要な機能のひとつは完璧なカスタマイズ機能です。要素を表示あるいは非表示にすることで思い通りに各パートを構成することができることを意味します。要素のヘッダをクリックしながらドラッグし要素をどこにでもドロップすることで同じ列の中でこの要素を再配置することさえ可能です。システムは、ボックスから使用する準備ができているいくつかのウィジェットを持っています。しかし、ダッシュボード画面のモジュール設計は、カスタム・ウィジェットを簡単に統合する準備ができています。

この画面のコンテンツは 2 つのメインカラムに整列され、その左カラムで各状態 (リマインダ、エスカレーション、新規、対応中といった) で分類されたチケットに関する情報を見ることができます。この各ウィジェットでは、他のフィルターの中で、アクセスすることを認められるチケット、ロックしたチケット、および他のフィルタ中に担当者が定義したキューにあるチケットのをすべて見るために結果をフィルタすることができます。このカラムには他の種類のウィジェットもあり、それらすべてを以下の述べます。

左列 ダッシュボード・ウィジェット

- チケット・リスト・ウィジェット

  このカテゴリーの下のウィジェットは全体的な同じ挙動、ルック・アンド・フィールを共有します。このウィジェットは決定した状態のチケットのリストを表示します。チケット数は各リスト・ページに表示します。リスト・ページは、ウィジェット・オプション(あなたがウィジェットの右上部上にマウスポインタを置くとそれらが現われます)で構成することができます。このウィジェットは次のフィルタを支援します:

  - 自身のロック・チケット

    ログインした担当者がロックされたチケット

  - 自身の監視チケット

    ログインした担当者がウオッチリストを持っているチケット。それを表示するために Ticket::Watcher 設定をオンにすることが必要です。

  - 自身の責任者

    ログインしている担当者が責任者として設定されているチケット。Ticket::Responsible設定は、このフィルターを可視化できるようにオンにする必要があります。

  - 自身のキューにあるチケット

    このチケットは担当者が「マイ・キュー」として定義するキューにあります。

  - 全てのチケット

    担当者がアクセス権を持つ全てのチケット

  このウィジェットは：

  - リマインダ・チケット

    保留として設定され、かつ、リマインダ日を経過したチケット。

- エスカレーションされたチケット

  エスカレーション済のチケット

- 新規チケット

  状態「新規」を持つチケット

- 対応中チケット / 返答必要

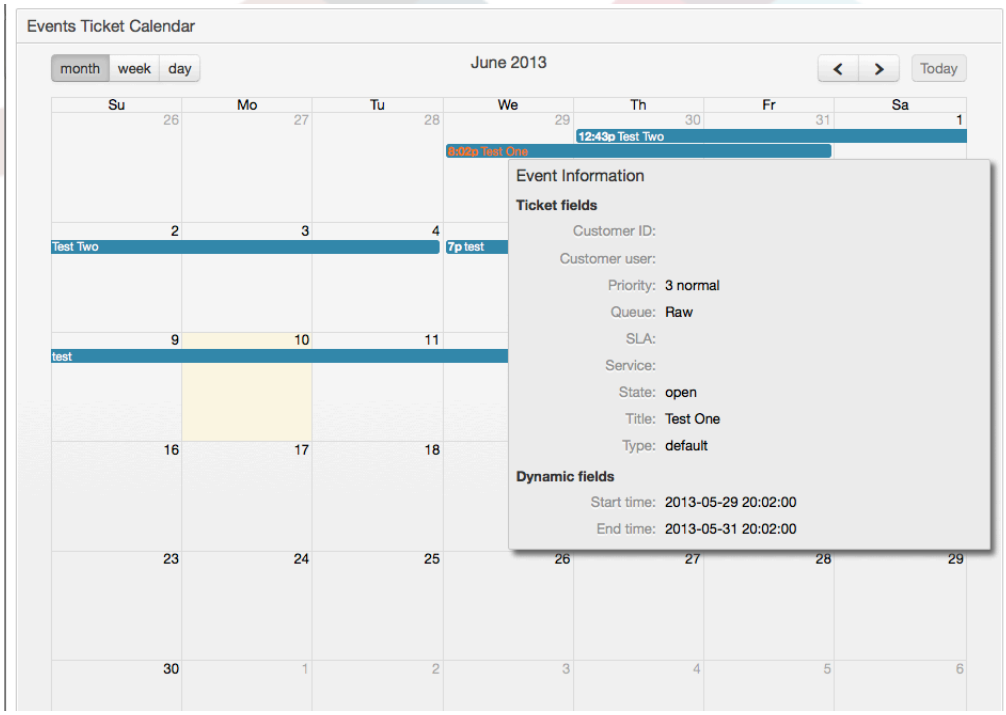
  状態「対応中」を持つチケットとそれらを対応する準備

- イベント・チケット・カレンダー

  カレンダ・イベントは新規チケットが作成された時、イベント・チケット・カレンダ機能が有効になります。それはチケット作成画面に表示する 2 つの新規フィールドを必要とし、ひとつはイベント開始時刻、他方は終了時刻用です。この 2 つの時刻がイベント期間を決定します。定義されます。

  このウィジェットは次のビューを含みます：月、週、日、及び左右の矢印を使用することでページからスクロールできる担当者です。

  直前にウィジェットは十分に有効になっていないと述べたようにチケット用の「Date/Time」の動的フィールドのいくつかはシステムに追加し、（管理者パネルにある動的フィールドリンク経由で）それらをこのウィジェットのためにSysConfigで設定すべきです。両方の動的フィールドはチケット生成画面に表示されるように構成されるべきです。それらは、カレンダのイベント（開始及び終了時間）用の時間枠について記述するために、チケット生成あるいは他のチケット・アクション画面(例えば自由フィールド)中に満たされるべきです。チケット・ズーム画面は、それが必要であると考えた場合、動的フィールドも示すように構成されるかもしれません。



  このウィジェットのためのさらなる構成は、SysConfig中の"Frontend::Agent::Dashboard::EventsTicketCalendar" サブグループの下で見つけられるかもしれません。

- カレンダー幅

カレンダの幅をパーセントで定義します。デフォルトは95%です。

- DynamicFieldStartTime

  開始時間の動的フィールド名を定義します。

- DynamicFieldEndTime

  終了時間の動的フィールド名を定義します。

- キュー

  この設定で明記されたキュー上のチケットのみが、カレンダ・ビューで検討されます。

- DynamicFieldsForEvents

  カレンダ・イベント・オーバーレイ画面に表示する動的フィールドを定義します。

- TicketFieldsForEvents

  カレンダ・イベント・オーバーレイ画面に表示するチケット属性を定義します。

- チケット・キュー・外観

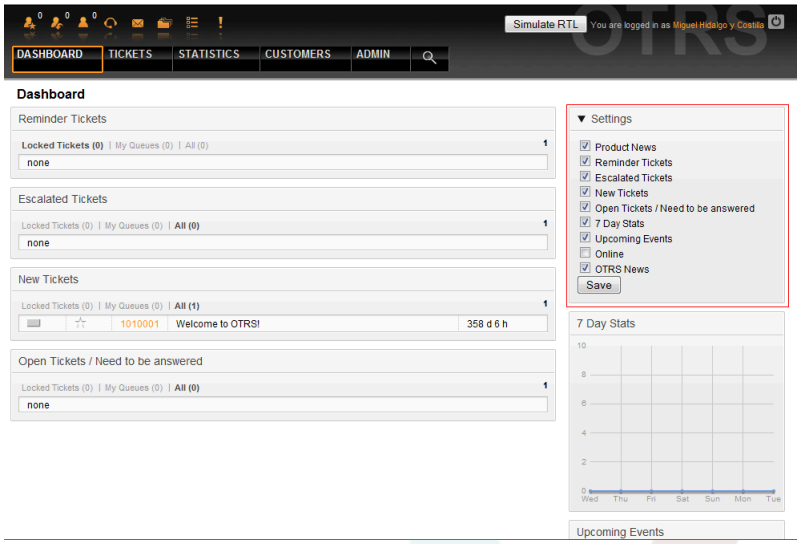
  このウィジェットは行がキューを表し列がチケット状態を表しているチケット・カウント・マトリックスの中で表示します。各セルには、特定のキュー上に属する定義された状態上のチケット数が表示されます。

  ウィジェットは、集計行および集計列も表示します。集計行は、すべての示されたキュー上の各状態のチケットの合計を表します。一方、集計列はすべての示された状態上で各キューのチケットの合計を表します。

  表示されるキューと状態はSysConfigで変更することができます

  チケット・カウント数のどれかをクリックすると検索結果ページが開き、担当者にそれらのより詳細な概要を得させます。

右列に特別なウィジェットが位置しており、非表示あるいは表示したいウィジェットをコントロールすることができます。これがウィジェットの設定です。セクションを拡張するヘッダをクリックします。すると下図のように全ての利用可能なウィジェットが見えます。各ウィジェット名にはチェックボックがあり、ダッシュボードでのウィジェットの可視性を定義するためにこのチェックボックスを使用します（チェックが外れているウィジェットは表示されません）。可視性オプションを定義したら、変更を格納するために「保存」をクリックします。このセクションはスクリーンに固定されるので、それをドラッグ＆ドロップができないし閉じることもできないということです。



図：ダッシュボード設定の編集

右列ダッシュボード・ウィジェット

- 7日統計

  それは、2行のラインを含んでいる過去7日にわたるチケット活動のグラフを示します。通常青い色である第 1 行は、日次作成チケット数を表示し、通常オレンジ色の第 2 行は日次完了チケットを表示します。

- 次回のイベント

  エスカレーションするためあるいは既にエスカレーションされた至急のチケットはここにリストされます。このウィジェットからの情報は大変役立ちます。なぜなら、注意喚起する必要のあるチケットを知る機会を持つからです。そうしてどのチケットに労力を傾注したいのか、優先度を置きたいのか、単に次の予定をチェックするのか、を決定することができます。

- OTRSニュース

  OTRSアクティビティの完全なリストと新製品のリリースやパッチに関する大変重要な情報

- オンライン

  ここで、現在システムにログインしている担当者に関する要約が示されます。さらにそれは、オンラインの顧客用セクションを含んでいます。このウィジェットは通常非表示であることに注目してください。それは前述の設定ウィジェットを使用して表示することができます。

7. キューとは何か？

多くのメールシステムでは、全てのメッセージが格納される受信ファイルに流し込むのが普通です。新しいメッセージはその受信ファイルの最後に追加されます。メール・クライアント・プログラムはメールの読み書きに使用され、この受信ファイルを読み、ユーザにコンテンツを表示します。

多くのメッセージを格納することもできるので、OTRSでのキューは多少受信ボックスと比較できます。キューにはさらに受信ボックスのメール・ファイル以上の特徴があります。OTRS担当者あるいはユーザとして、チケットがどのキューに格納されているかを記憶しておく必要があります。担当者は、キューにあるチケットを開き編集することができ、あるキューから別のキューに移動することもできます。しかし、なぜチケットを移動させるのでしょうか。

それについてより実践的な説明するために、example of a ticket systemで述べたマックスの会社の例を思い出しましょう。マックスは彼のチームがビデオテープレコーダーを買う会社取引先のサポートをよりよく管理することを可能にするためにOTRSをインストールしました。

全てのリクエストを持つ1つのキューは、この状況にとって十分です。しかし、マックスがさらにいつかDVDレコーダーを売ることを決定後は、顧客は、ビデオテープレコーダーだけでなく新製品に関する質問をします。より多くのEメールがマックスのOTRSの単一のキューに入り、起こっていることの明瞭な事態を把握することは難しくなります。

マックスは、サポート・システムを再構成することを決定し、2つの新しいキューを追加します。したがって、これで3つのキューが使用されていることになります。チケットシステムに到着する新しいメッセージは、古いキュー「未対応(raw)」へ格納されます。2つの新しいキューのうち、1つ「ビデオテープレコーダー」はもっぱらビデオテープレコーダーのリクエスト用、もう一方の「dvdレコーダー」はもっぱらdvdレコーダーのリクエスト用です。

マックスはサンドラに未対応キューを注視し、顧客の要求に応じてメッセージを「ビデオレコーダ」キューまたは「dvdレコーダ」キューにソート（送る）します。

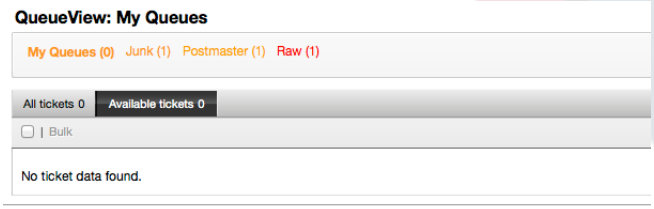
OTRSは、ユーザ、グループおよび役割のためのアクセス管理をサポートします。いくつかのユーザアカウントにのみアクセス可能なキューを設定することも簡単です。マックスは、さらにフィルタ・ルールで、リクエストを異なるキューに入れる別の方法を使用することができました。そうでなければ、2つの異なるメールアドレスが使用される場合、サンドラは単に自動的に送信できない2つの他のキューへそれらの電子メールを送信しなければなりません。

異なるキューへの受信メッセージをソートすることは、サポート・システムを組み立てて、かつ整然としていておくのを助けます。担当者はキュー上の異なるアクセス権を持つ異なるグループへ配置されるので、システムはさらに最適化することができます。キューは、ワークフロー・プロセスを定義するかあるいは会社の構造を作るために使用することができます。マックスは例えば、「要求、オファー、注文、請求」等を含む「販売」と呼ばれる別のキューを実行することができました。マックスにとってそのようなキュー構造は、注文処理を最適化するのを助けることができました。

キューの適切なデザインを通してのような改善されたシステム構造により、重要な時間および原価の削減につなげることができます。キューは、会社の中のプロセスを最適化するのを支援することができる。

8. キュー概要とは何か？

キュー概要は（チケットが表示され、読み書き権限を持つユーザに対する）全てのキューの一覧を提供します。



キュー概要は、OTRSの日常業務に対して様々なオプションを提供します。まずはじめに、マイキューです。担当者選択（選好）であるいは担当者を管理する場合、内部で作業するために指定された担当者のためにキューのセットは定義可能です。Ticket -> Queue View メニューにアクセスする時、すべてのチケットがこのデフォルト・ビューに表示されます。

キュー・ビューによって提供される 2 番目のオプションは、個別のキューと作業すべきチケット含むサブ・キューにドリルダウンするナビゲーションです。

両方のビュータイプでユーザはさらに追加の能力を持ちます。全てのアンロック・チケット、あるいは利用可能なチケットを見ることを選択できます。チケットは、キュー・ビューで見える可視状態タイプの一つでなければなりません。デフォルトでは「対応中、新規、保留、リマインダ、保留-自動」があります。

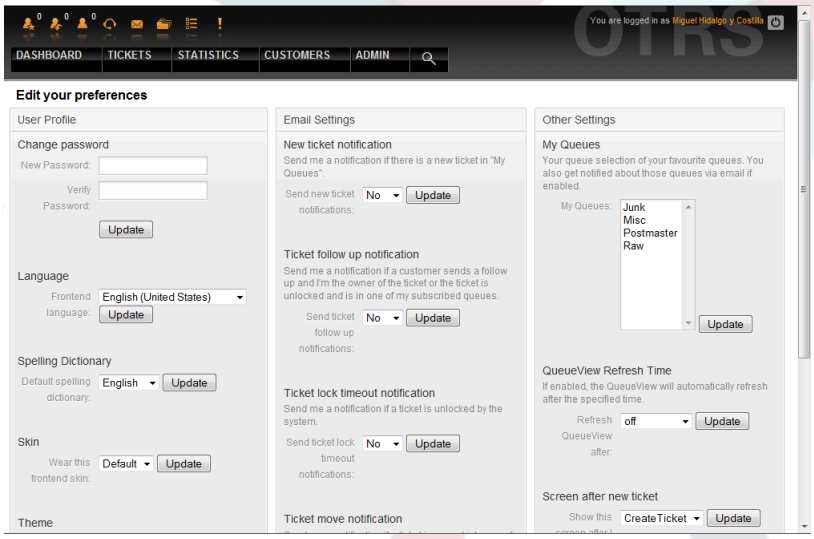
ユーザ補助のための視覚アラームがあります。

視覚アラーム

- ハイライト 時間 1：未対応のチケットを含んでいるハイライトされたキューに対して、数分(第1のレベル)の時間をセットします。「未対応」キューの中で上部に表示されます。
- ハイライト 時間 2：未対応のチケットを含んでいるハイライトされたキューに対して、数分(第2のレベル)の時間をセットします。「ポストマスター」キューの中で上部に表示されます。
- 点滅 ：最も古いチケットを含んでいるキューの明滅するメカニズムを活性化します。すべてのブラウザでサポートされているわけではありません。その場合「ジャンク」キューの上部に赤色で見えます。
- 太字：現在のキューは「マイキュー」の上部に太字で見えます。

9. ユーザ選択（選好）

OTRSユーザ（顧客、担当者、OTRS管理者としての）は、必要に応じて自身のアカウントの選択（選好）を構成できます。担当者はウェブインターフェイスの上部右隅にあるログイン名をクリックして構成画面にアクセスできます。顧客は「選択（選好）」リンクをクリックしなければなりません（下図参照）。



図：担当者の個人選択（選好）

担当者は選択（選好）から 3 つの異なるカテゴリーを構成できます。ユーザプロファイル、Eメール設定及びその他設定です。デフォルトの可能性は：

ユーザ・プロファイル

- 現在のパスワードを変更
- インターフェイス言語を変更
- フロントエンド・スキンを変更
- フロントエンド・テーマを変更

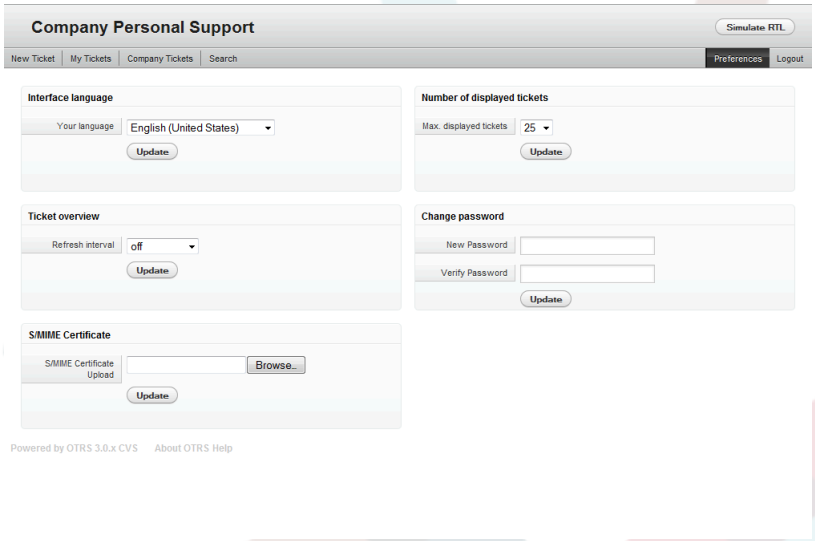
- 業務外時間を活性化し構成

Eメール設定

- 担当者へEメール通知をきっかけとするイベントを選択

その他の設定

- 「マイキュー」でモニターしたいキューを選択
- キュー・ビュー用更新期間を設定
- チケット作成後表示する画面を設定



図：顧客の個人選択（選好）

顧客はウェブ・インターフェース言語を選択し、チケット概要用の更新間隔をセットし、表示するチケット最大数を選択できます。また新しいパスワードをセットすることも可能です。

第4章 管理

1. OTRSの管理者エリア

1.1. 基礎

次のシステム構成セッティングは、OTRS管理者が、OTRSウェブインターフェースの管理者ページへのアクセスをすることで利用可能です。それにより可能なことは、担当者、顧客、そしてキューの追加。チケットとEメールの構成、FAQやITSMなどさらに多くの追加のパッケージをインストールすることです。

管理者グループのメンバーである担当者は、ナビゲーション・バー中の管理者リンクのクリックにより管理者エリアにアクセスすることができます(下記の図を参照)。十分に高いアクセス権のない担当者はこのリンクにアクセスすることができません。

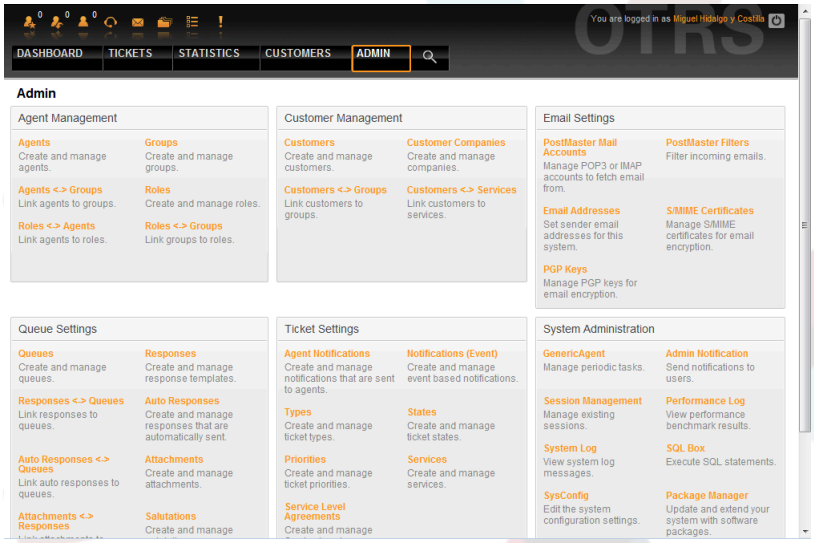


図: *OTRS*管理者画面。

1.2. 担当者、グループおよび役割

1.2.1. 担当者

担当者のリンクをクリックし、OTRSの担当者管理画面にアクセスします(下記の図を参照)。管理者は担当者アカウントを追加、変更、非活性化することができます。更に、管理者は個々の担当者のインターフェースのための言語と通知に関する構成を含む担当者選好を管理することができます。

注記

OTRSの担当者アカウントは非活性化されますが、ここでは削除はされません。非活性化は無効または無効-暫定のフラグを有効にセットするで実施されます。

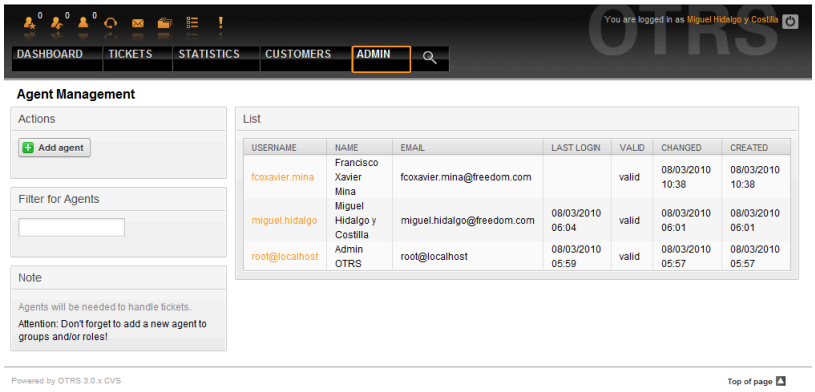


図:担当者管理。

担当者を登録するには図の中で示されるように、「担当者追加」ボタン上のクリックし、必要なデータを入力し、画面の最下部の送信（Submit）ボタンを押します。

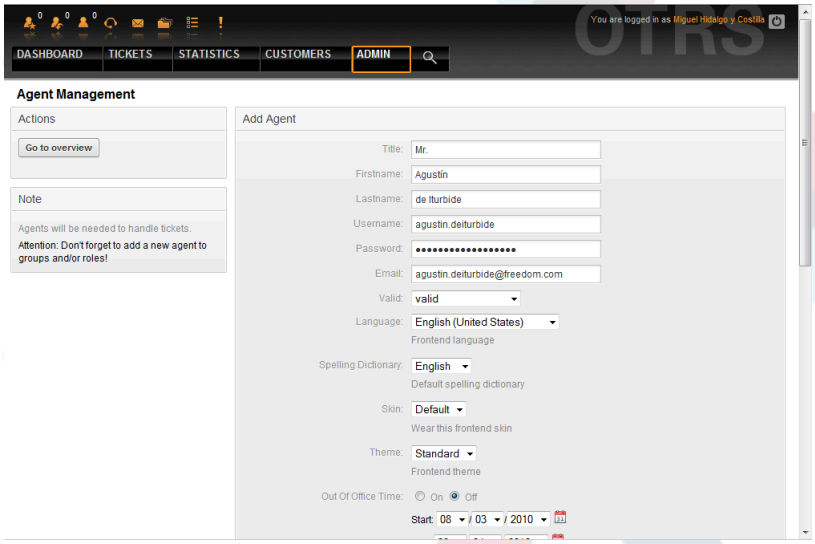


図:新規担当者の追加。

新規担当者アカウントが作成された後、担当者に1つ以上のグループあるいは役割のメンバーを作るべきです。グループと役割に関する情報は本章のグループ と役割のセクションにおいて利用可能です。

1.2.2. グループ

すべての担当者アカウントは少なくとも1つのグループあるいは役割に属するべきです。新規インストールでは、テーブル5-1に示されるように、利用可能な3つの事前定義されたグループがあります。

表**4.1** デフォルト・グループは新規の**OTRS**インストールで利用可能です

グループ	摘要
admin	システムで管理業務を実行することができます。

グループ	摘要
stats	OTRSのstatsモジュールにアクセスし、統計を生成する資格があります。
users	担当者は読み書きの権限を持って、このグループに属し、チケットシステムのすべての機能にアクセスすることができます

注記

新規OTERSインストレーションでは、最初はユーザグループにメンバーは存在しません。担当者'root@localhost'はデフォルトでは管理者及び状態グループに属します。

管理者エリアのグループリンクをクリックするとグループ管理ページ(下記の図を参照)にアクセスすることができます。

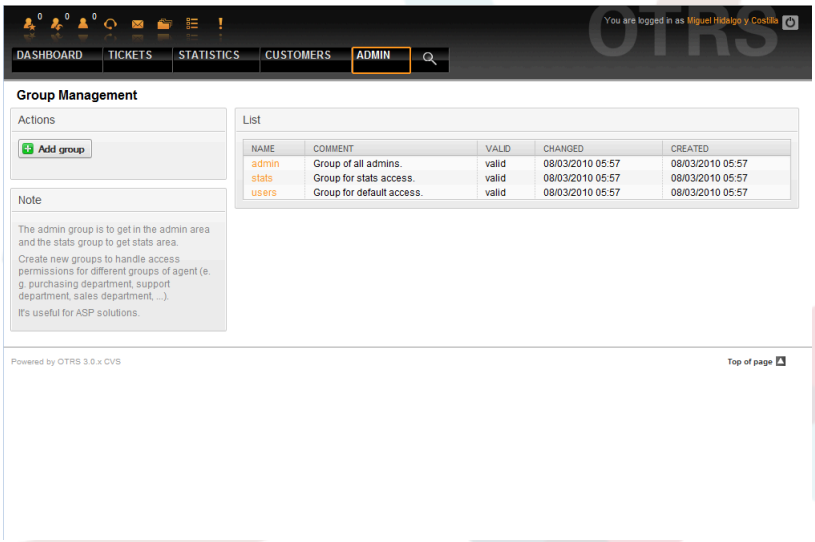


図:グループ管理。

注記

担当者でそうであるように、OTRSグループは非活性化することができますが削除はできません。非活性化は無効または無効-暫定のフラグを有効にセットするで実施されます。

担当者をグループに加えるかあるいはグループに属する担当者を変更するために、管理者ページから担当者リンク<-> グループを使用することができます(下記の図を参照)。

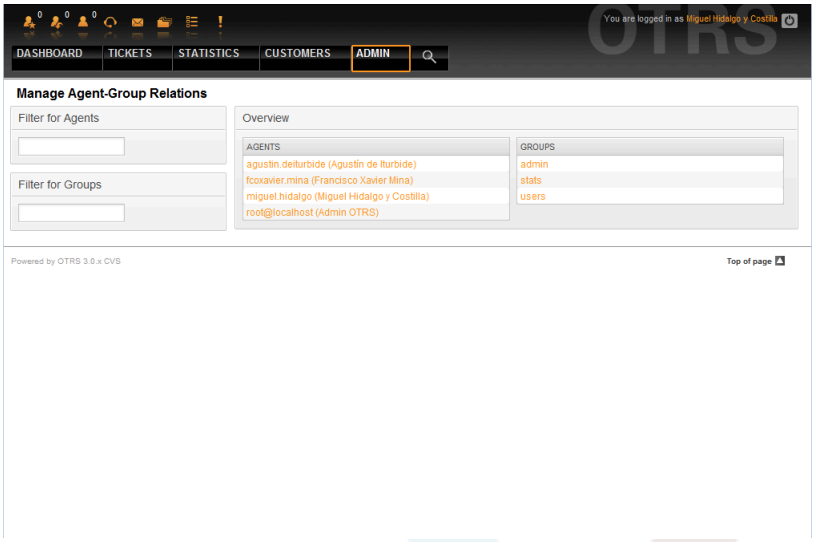


図:グループ管理。

システムでのすべてのグループおよび担当者の概要はこのページに表示されます。さらに、特定のエンティティ（実体）を検索するために利用可能なフィルタを使用することができます。担当者がメンバーであるグループを変更したい場合は、担当者の名前をクリックしてください(下記の図を参照)。グループに関連付けされた担当者を変更するためには、編集したいグループをクリックしてください(下記の図を参照)。

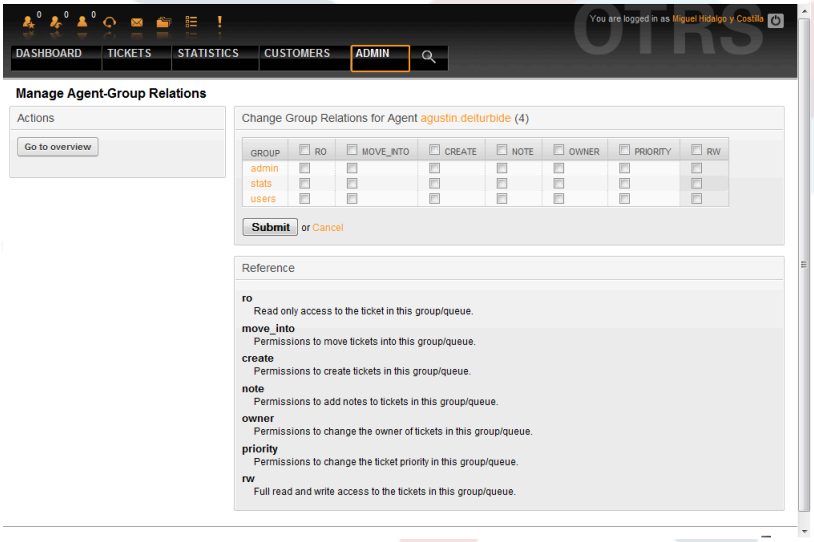


図:担当者が属するグループを変更。

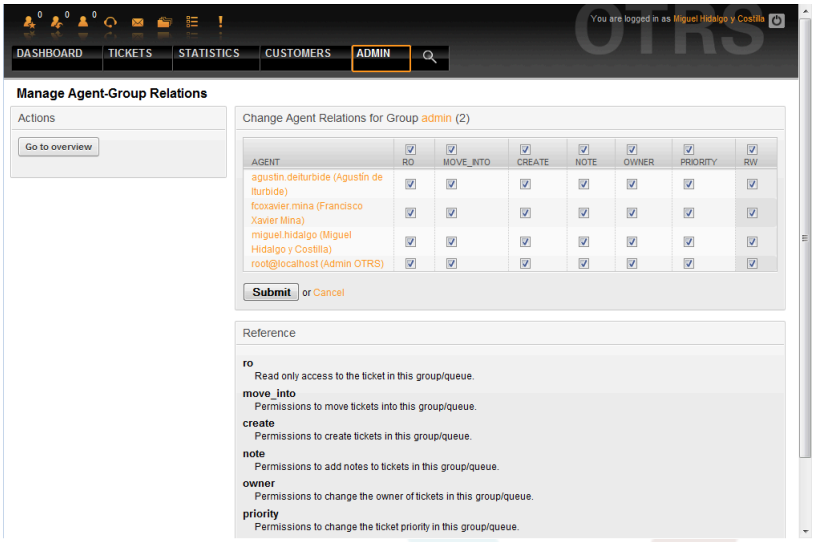


図:特定のグループに属する担当者を変更。

各グループはそれに関連した権限のセットを保持しています。そして各グループメンバー(担当者)は各々自分のためにこれらの権限の組み合わせを持つことができます。許可/権限のリストはテーブル5-2を御覧ください。

表4.2 OTRSグループと関連した権限

権限	摘要
ro	このグループのチケット、エントリー、キューへのRead onlyのアクセス
move into	このグループのエリア、キュー間におけるエントリーあるいはチケットの移動権限
create	このグループのエリアまたはキュー間におけるエントリーあるいはチケットの作成権限
owner	このグループのエリアまたはキュー間におけるエントリーあるいはチケットのオーナーの更新権限
優先度 (priority)	このグループのエリアまたはキュー間におけるエントリーあるいはチケットの優先度の変更権限
rw	このグループのエリアまたはキュー間におけるエントリーあるいはチケットのフルのRead/Write権限

注記

By default, the QueueView only lists tickets in queues that an agent has *rw* access to, i.e., the tickets the agent needs to work on. If you want to change this behaviour, you can set Ticket::Frontend::AgentTicketQueue###ViewAllPossibleTickets to *Yes*.

すべての利用可能なパーミションはデフォルトで表示されるとは限りません。これらの追加のパーミションは加えることができます。

表4.3 追加の許可グループ

権限	摘要
stats	stats(統計)ページにアクセス権を与えます。

権限	摘要
bounce (返送)	Eメールメッセージを送り返す権限 (チケットズームの返送ボタンで) 。
compose (作成)	チケットに対する回答を作成する権限。
customer (顧客)	チケットの顧客を変更する権限。
forward （ 転送 ）	メッセージを転送する権限(転送ボタンで)。
pending （ 保留 ）	チケットを保留にする権限。
phone （ 電話 ）	チケットに通話を追加する権限。
responsible （ 責任者 ）	チケットに対する責任担当者を変更する権限。

注記

権限の変更はSystem::Permissionから変更可能です

1.2.3. 役割

Roles:"役割"は、強力な機能で、それにより非常にシンプルかつ迅速に多くの担当者のアクセス権を管理できます。それらは、多くの担当者、グループおよびキューを抱える大きく複雑なサポート・システムで特に役立ちます。下記の例でどのようなケースでご利用いただけば良いかを説明します。

あなたのシステムが100人の担当者を持つと仮定します。90 of them with access to a single queue called "support" where all support requests are handled. 単一のキューにアクセスを持つ100人中90人は、全てのサポート依頼を取り扱う「サポート」と呼ばれるます。「サポート」キューは複数の副キューを含みます。残りの10人の担当者は、システムの全てのキューへアクセスする許可を持っています。この10人の担当者は、チケットを発行し、未対応のキューを見て、「ジャンク」キューへスパム・メッセージを移動させます。

その会社は、何らかの製品を販売する新規部署を開設します。注文依頼、受注、注文確認あるいは支払などの処理が必要となるでしょう。そして、会社の担当者のうちの数人は、OTRSを使用して、これをすることになります。異なる担当者は、生成される新規のキューへのアクセスを得なければいけません。

個々の担当者のためにアクセス権を手動で変更するのに長い時間がかかるので、異なるアクセスレベルを定義する”役割”を作成することができます。その後、担当者は、自動的に修正済のアクセス権を持つことにより、1つ以上の役割に加えることができます。新規担当者アカウントが作成される場合、さらに、このアカウントを1つ以上の役割に加えることは可能です。

注記

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!

管理ページにある 役割 リンクをクリックすることで、役割管理セクション(下図参照)にアクセスできます。

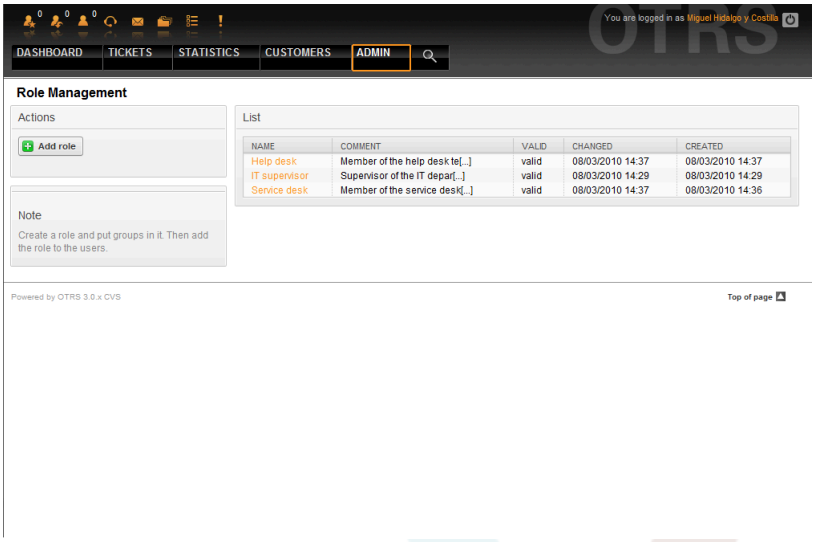


図:　役割管理。

注記

担当者とグループでそうであったように、いったん作成された役割は、非活性化することができるが削除することができません。非活性化するためには、有効オプションで「無効」あるいは「無効-暫定」にセットするします。

システムでのすべての役割の概要が表示されます。役割設定を編集するためには、役割の名前をクリックしてください。新規のOTRSインストレーションでは、デフォルトで定義された役割はありません。役割を登録するためには、「役割追加」ボタンをクリックして、必要とされるデータを入力し、それを送信(Submit)して下さい(下記の図を参照)。

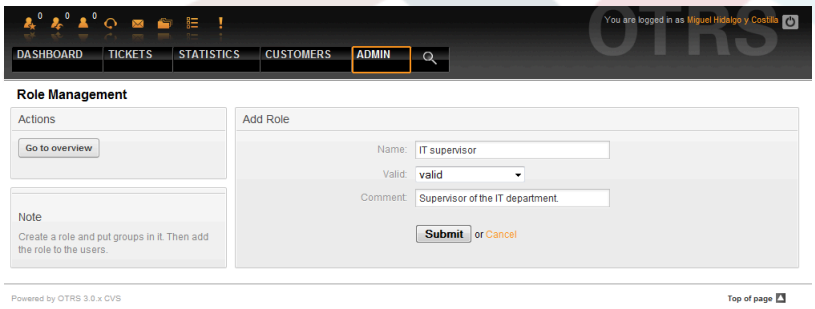
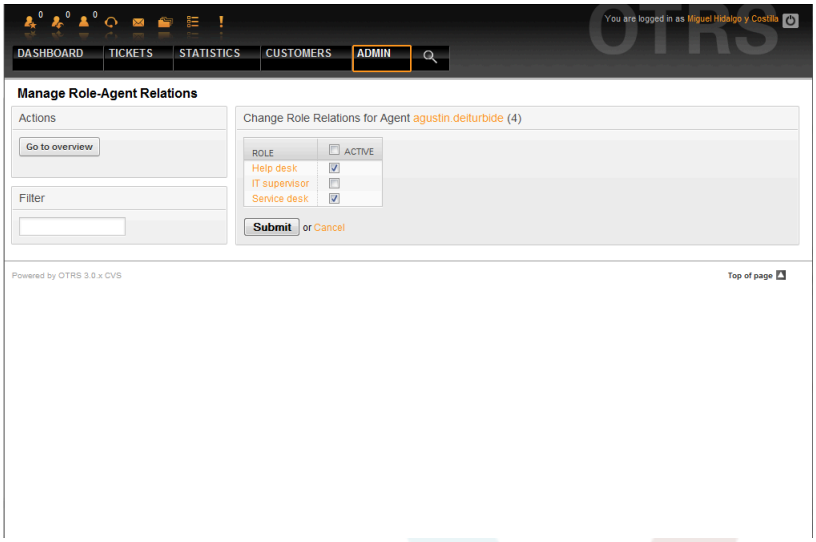
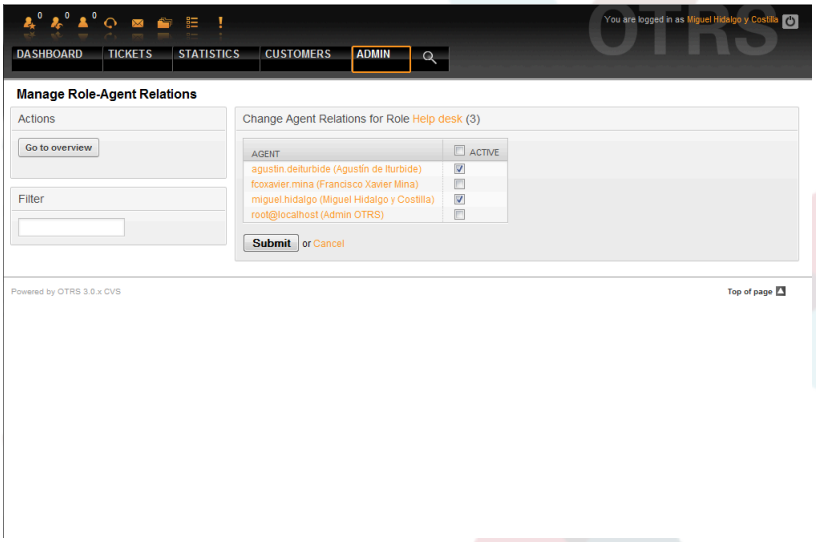


図:役割の新規追加。

システムですべての役割および担当者の概要を得るためには、管理者ページの役割リンク<-> 担当者をクリックして下さい。さらに、特定の要素を見つけるためにフィルタを使用することができます。担当者に関連付けられた役割を変更したい場合は、担当者の名前をクリックしてください(下記の図を参照)。役割に関連付けられた担当者を変更するためには、あなたが編集したい役割をクリックしてください(下記の図を参照)。

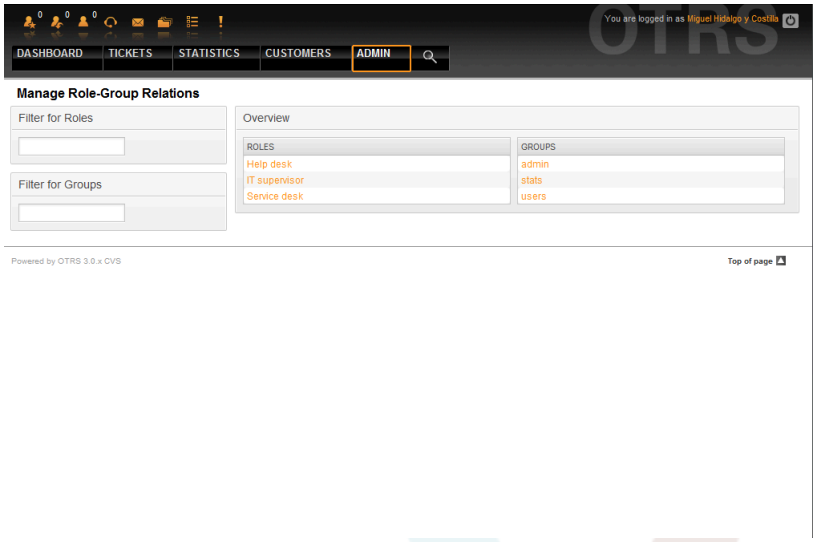


図：担当者と関連付けられた役割の変更



図：特別な役割と関連付けられた担当者の変更

システムでのすべての役割およびグループの概要を得るためには、管理者ページのリンク・役割<->グループをクリックしてください。図で表示されたと同様の画面が見えることでしょう。さらに、特定のエンティティ（実体）を見つけるためにフィルタを使用することができます。



役割グループ関係の管理

役割用の異なるアクセス権を定義するためには、役割またはグループの名前をクリックしてください(図5.13および5.14以下にそれぞれ参照)。

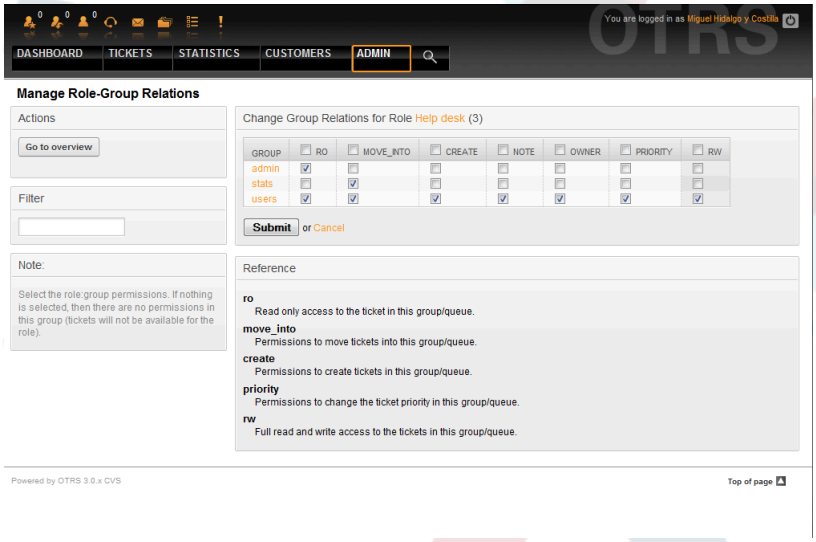


図: 役割のためのグループ関係の変更

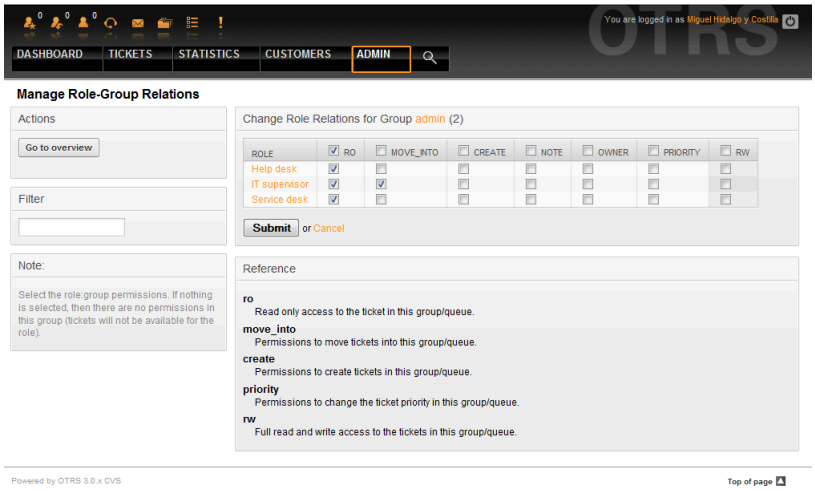


図:　グループ用役割関係の変更

1.3. 顧客および顧客グループ

1.3.1. 顧客

OTRSは、異なるタイプのユーザをサポートします。顧客リンク(ナビゲーション・バー、あるいは管理者ページから)を使用すると、顧客(下記の図を参照)のアカウントを管理することができます。顧客・インターフェース(customer.pl)経由でシステムにログインすることができます。このインターフェースを通って、あなたの顧客はチケットを作成することができるだけでなく、更新するために過去のチケットをレビューすることもできます。顧客はシステムでのチケット履歴を必要としていることを知ることは重要です。

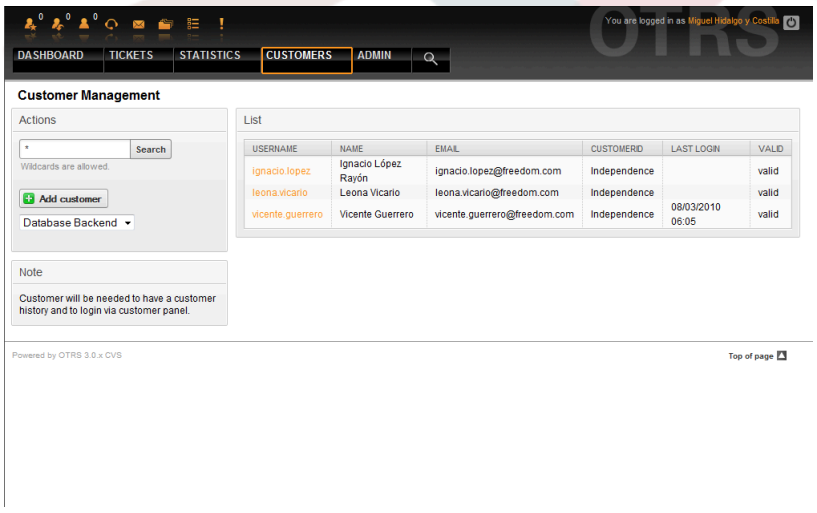


図: 顧客管理

登録した顧客を検索し顧客名をクリックすることで設定を編集できます。顧客のバックエンドを変更することもできます。詳細は外部バックエンドの章を参照下さい。

新規顧客アカウントを作成するためには、「顧客追加」ボタンをクリックしてください(下記の図を参照)。フィールドのうちのいくつかは入力必須項目です。つまり、フィールドには値の入力が必要です。したがって、必須入力項目に入力漏れがあれば、赤でハイライトされます。

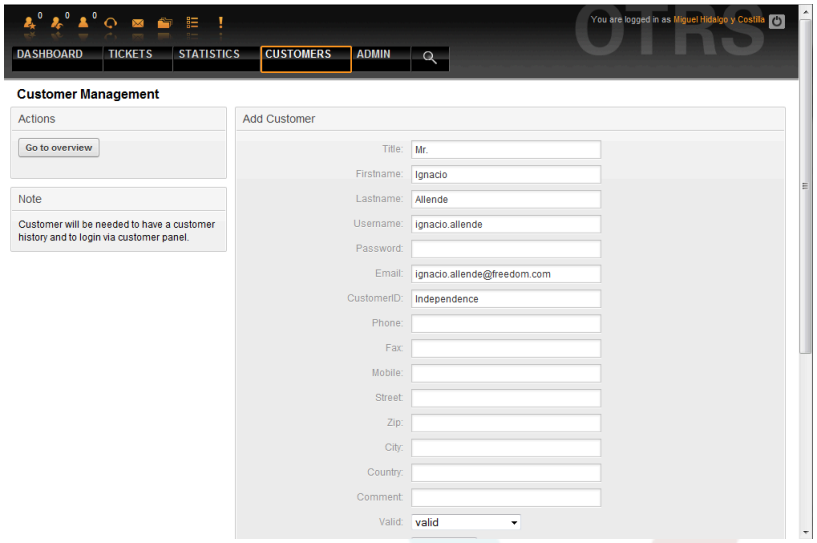


図: 顧客追加

顧客はそれらのユーザー名とパスワードの提供されることでシステムにアクセスすることができます。CustomerIDはユーザと関連チケットを識別するためにシステムによって必要です。メール・アドレスがユニークな値であるので、それはIDとして使用することができます。

注記

担当者、グループおよび役割でそうであったように、顧客は、システムから削除されません。単に、有効オプションが無効または無効-暫定に設定されることで非活性化されるだけです。

1.3.2. 顧客グループ

顧客ユーザもグループに加えることができます。1つあるいは少数のキューへのアクセスと同じ会社の顧客を加えたければ、役立つでしょう。まず、グループ管理モジュールによって、あなたの顧客が属するグループを作ってください。次に、キューを加えて、キューに新規グループを選んでください。

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.

リンク"顧客<-> グループ" によって、異なるグループ(下記の図を参照)に属する顧客を管理できます。

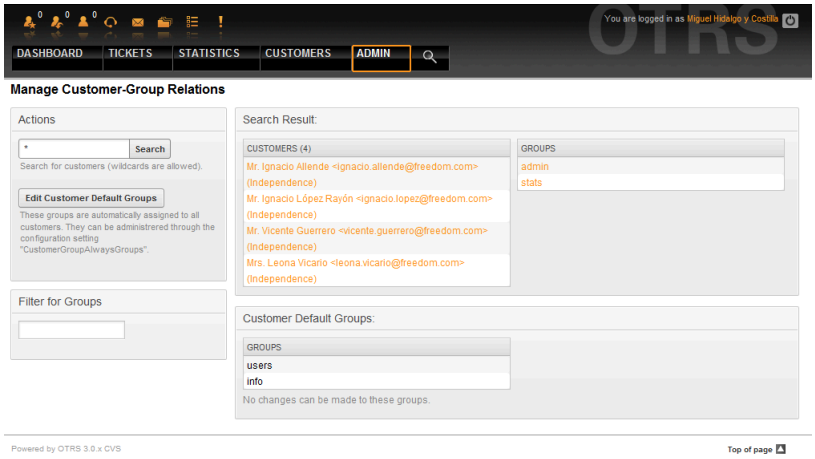


図:顧客グループ関係管理

異なるグループを定義するために、顧客もその一部分であるべきです。対応する顧客名やグループをクリックします(図5.16および5.17以下にそれぞれ参照) 。

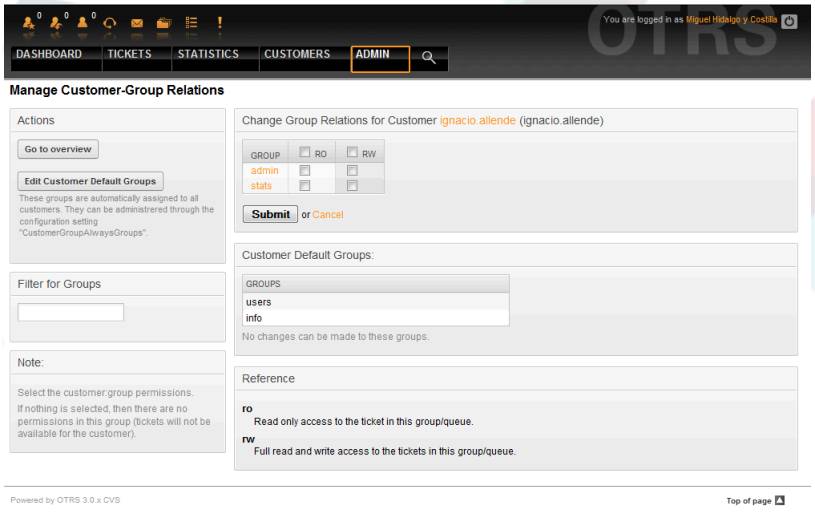


図:顧客用グループ関係の変更

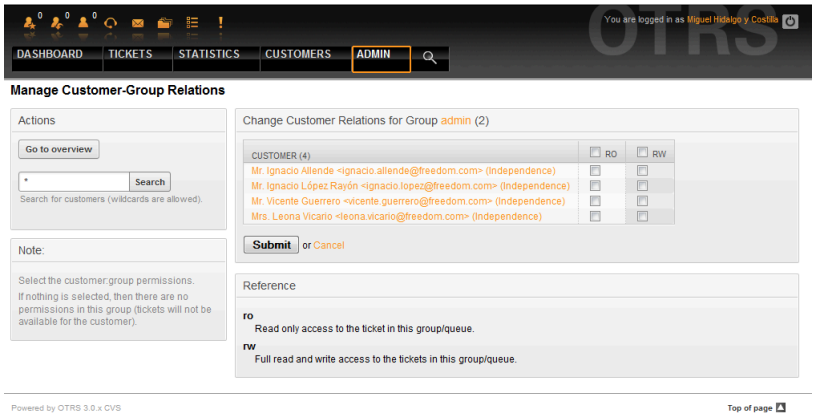


図:グループのための顧客関係の変更

1.4. キュー

管理者ページをリンク「キュー」をクリックすると、システムのキューを管理することができます(下記の図を参照)。新規OTRS装置では、4つのデフォルト・キューがあります:「未対応」、「ジャンク」、「その他」と「ポストマスター」。フィルタ規則が定義されなければ、受信メッセージはすべて「未対応の」キューに格納されるでしょう。「ジャンク」キューはスパム・メッセージを格納するために使用することができます。

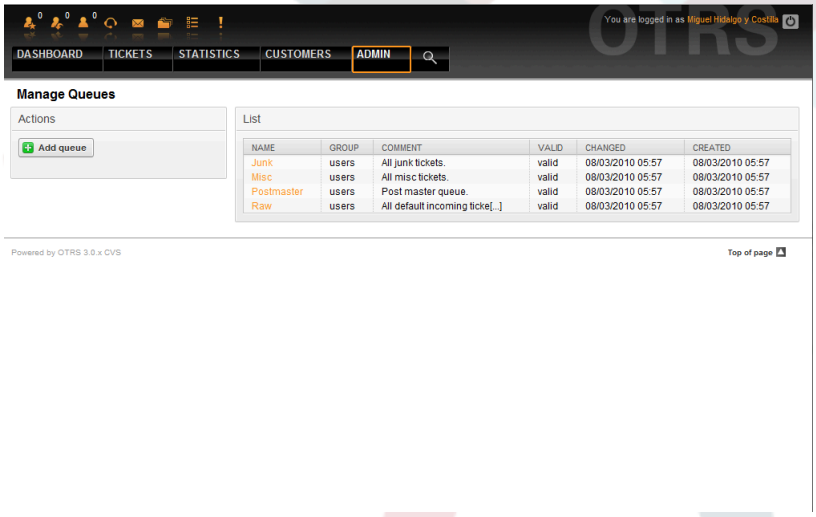


図:キュー管理

ここで、キュー(下記の図を参照)を加えて、それらを修正することができます。キューを使用するべきグループを指定することができます。さらに、既存のキューのサブキューとしてキューをセットすることができます。

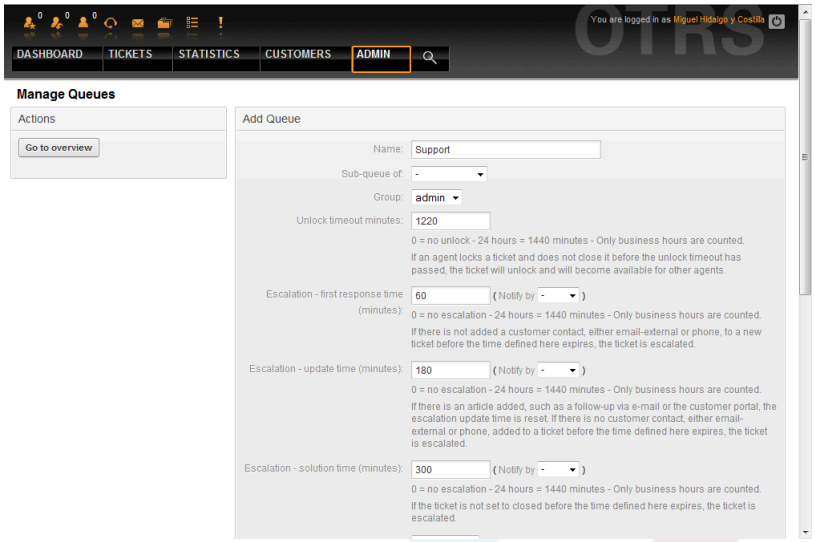


図:新規キューの追加

キュー用のアンロック・タイムアウトを定義することができます。担当者がチケットをロックし、アンロックタイムアウトが過ぎる前に、それを完了しなければ、チケットが自動的にアンロックされ、他の担当者が継続して利用可能にされます。

キュー・レベルに関連可能な3つのエスカレーション時間設定があります:

エスカレーション - 第1の応答時間

- チケットの生成の後、顧客といかなるコミュニケーションも無く、定義された時間が失効した場合、Eメールであれ電話であれエスカレーションされます。

エスカレーション - 更新時間

- Eメールあるいは顧客ポータル(それはチケットに記録される)のいずれかによって顧客フォローアップがある場合、エスカレーション更新時間はリセットされます。ここで定義された時間が失効するまでに顧客へのコンタクトがない場合、チケットはエスカレーションされます。

エスカレーション - 解決時間

- ここで定義された時間が失効する前にチケットが完了（成功）しなかったら、チケットはエスカレーションされます。

「フォローアップの後のチケット・ロック」により、以下を定義できます。１．旧チケット・オーナーに対して「ロック」状態にセットすべきチケット。２．一旦完了してて、再度対応中となったチケットこれにより以前にそのチケットを扱った担当者が確実にフォローアップ・チケットが処理することができます。

システム・アドレス用のパラメタは、このキューの送信チケットに使用されるメール・アドレスを特定します。Eメール返信のために、挨拶文と署名にキューを関連させることもできます。より多くの詳細情報に関しては、Eメール・アドレス、挨拶文および 署名のセクションを参照してください。

注記

担当者でそうであるように、グループ、顧客、キューは削除できません。単に有効オプションを無効にあるいは無効-暫定にする設定によって非活性化するだけです。

1.5. 挨拶文、署名、添付およびテンプレート

1.5.1. 挨拶文

挨拶文はテンプレート用のテキスト・モジュールです。キューに関するセクションに述べられているように、挨拶文は1つ以上のキューにリンクすることができます。挨拶文は、挨拶文がリンクされて、返信されているキューからのチケットのみに使用されます。システムの異なる挨拶文を管理するためにには、管理者エリアの「挨拶文」リンクを使用してください(下記の図を参照)。

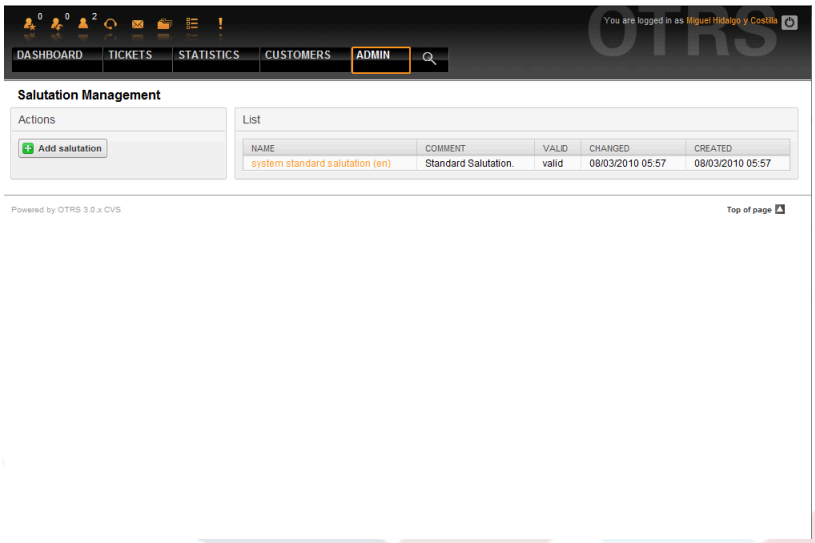


図:挨拶文管理

デフォルト・インストレーションの後、利用可能な1つの挨拶文が既にあります、「システム標準挨拶文(en)」

新規挨拶文を作成するには「挨拶文の追加」ボタンを押して、必要なデータを提供し、送信して(下記の図を参照)下さい

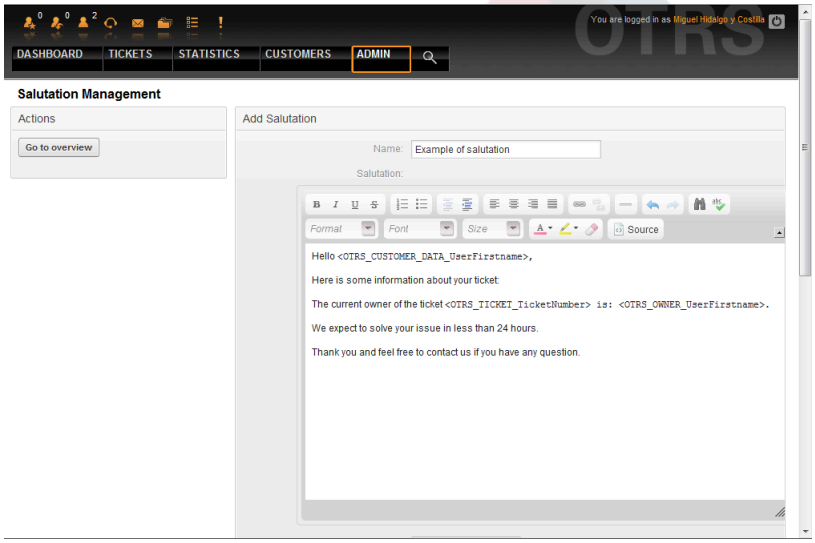


図:新規挨拶文の追加

挨拶文の中で変数を使用することは可能です。チケットに応答する時、変数名はそれらの値と取り替えられるでしょう。

テンプレートの中で使用することができる異なる変数は、挨拶文画面の下方部分にリストされます。例えばこれを使えば、チケット送信者の姓が変数<OTRS_LAST_NAME> が返信に含まれるでしょう。

注記

他のOTRSの実体のように、挨拶文は削除できません。有効オプションを無効または無効-暫定にすることで不活性化されるだけです。

1.5.2. 署名

テンプレート用の別のテキスト・モジュールは署名です。キューに関するセクションに述べられているように、署名はキューにリンクすることができます。署名が以前にキューにリンクされていれば、署名は単にテンプレート・テキストに追加されるだろうということに注意してください。管理者ページの「署名」リンクへのアクセスにより、システムでの署名を管理することができます(下記の図を参照)。

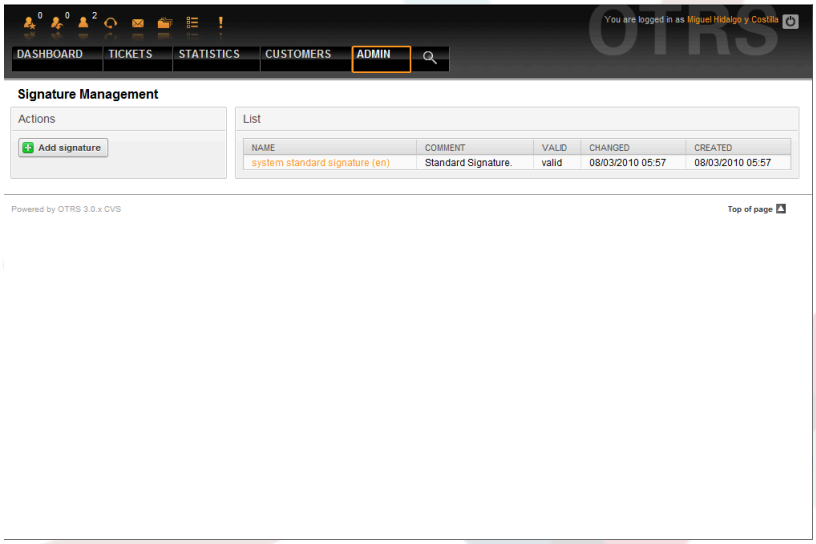


図: 署名管理

OTRSの新規インストール後、システムにはあらかじめ定義された署名：「システム標準署名(en)」が存在します。

新規署名を作成するために、「署名の追加」ボタンを押して下さい。必要とされるデータを提供し、送信します(下記の図を参照)。

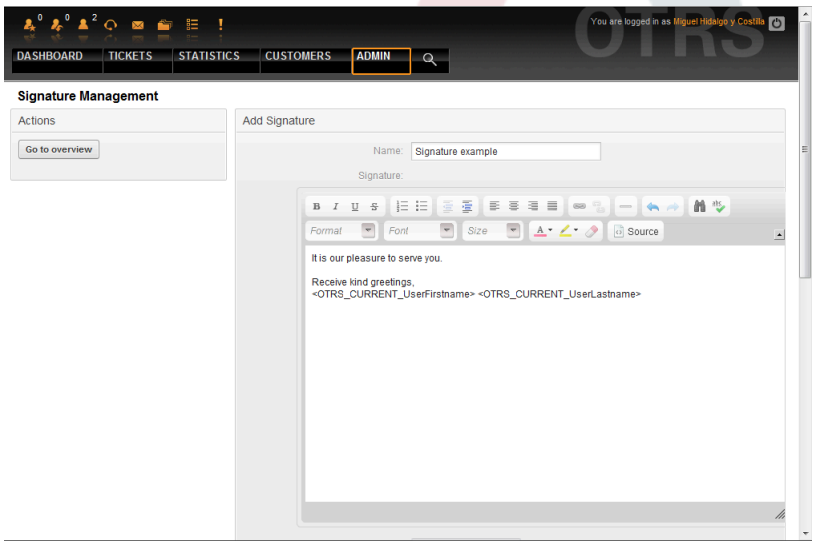


図:新規署名の追加

挨拶文のように、署名は、チケットに答える担当者の姓、名のような動的内容を含むことができます。またここで変数は、すべてのチケットに対して署名テキストのコンテンツを代替するために使用することができます。使う変数に関しては署名画面の下部をご覧ください。あなたが署名に例えば可変<OTRS_LAST_NAME>を含めば、チケットに答える担当者の姓は変数を代替します。

注記

挨拶文でもそうであるように、署名も削除できません。有効オプションを無効または無効-一時的に設定してのみ非活性化することができます。

1.5.3. 添付

さらに、任意に1つ以上の添付をテンプレートに加えることができます。テンプレートが選択されていれば、添付はメッセージ作成ウィンドウに付けられるでしょう。必要ならば、担当者は顧客のもとへそれを送る前に個々のテンプレートから添付を取り除くことができます。

管理者ページの「添付」リンクによって、システムのデータベースに添付を読み込むすることができます(下記の図を参照)。

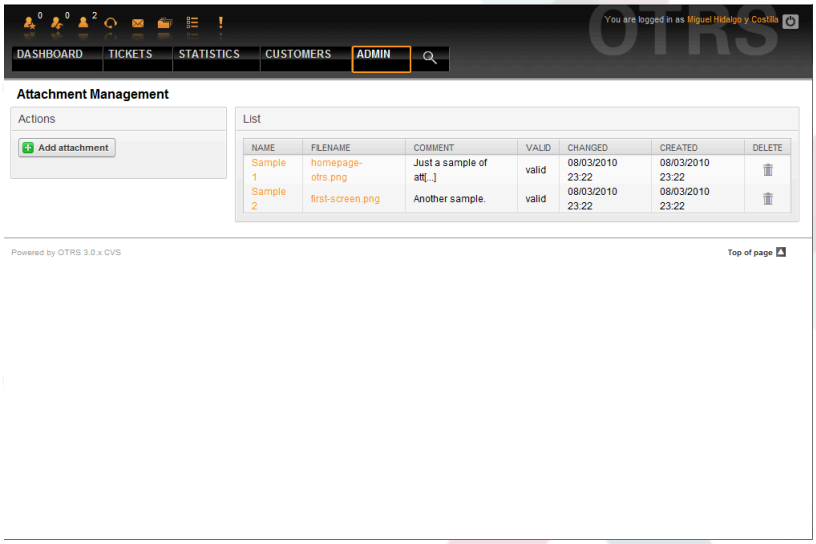


図:添付管理

新規添付を作成するために、「添付追加」ボタンを押し、必要なデータを提供し、送信します(下記の図を参照)。

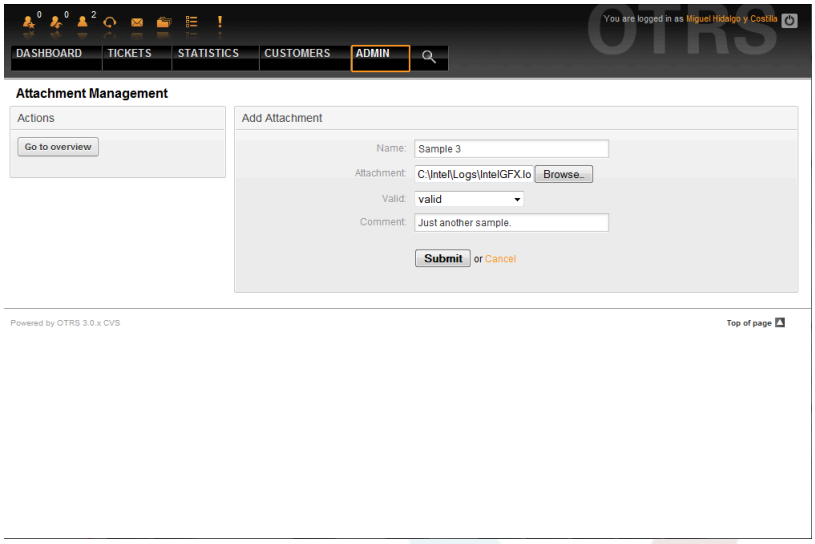


図:新規添付の追加

添付が格納される場合、1つ以上のテンプレートにリンクすることができます。管理者ページ(下記の図を参照)のリンク「添付<-> テンプレート」をクリックして下さい。

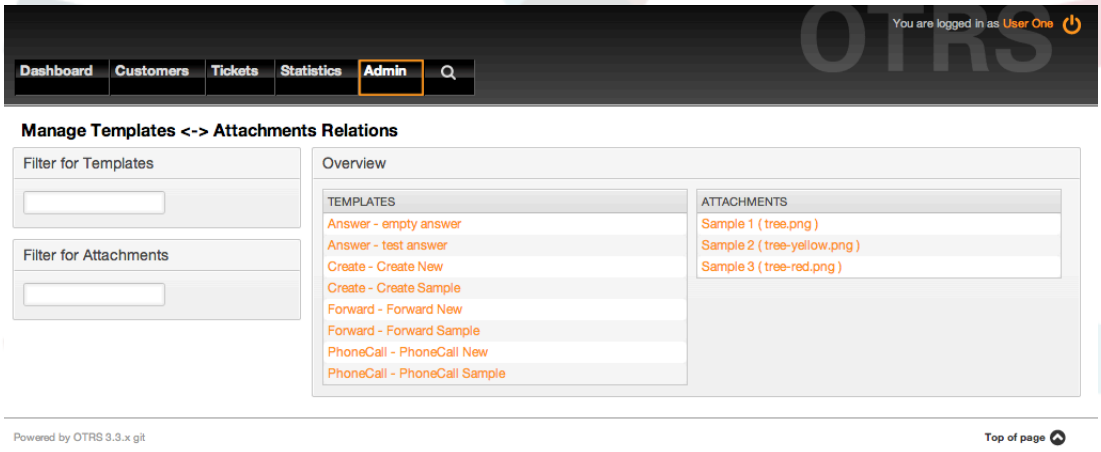


図:添付のテンプレートへのリンク

特定のテンプレートに、および逆に異なる添付を関連させるためには、対応するテンプレート名あるいは添付をクリックしてください(図5.27および5.28以下にそれぞれ参照)。

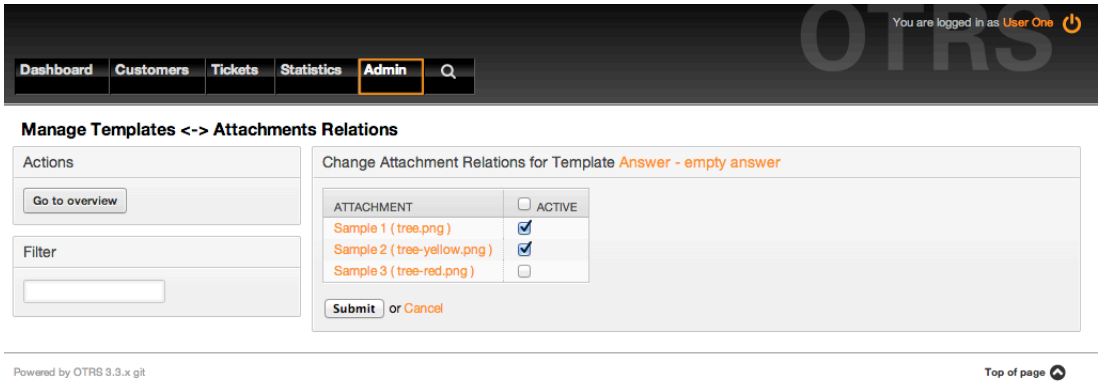


図:テンプレート用添付関連の変更

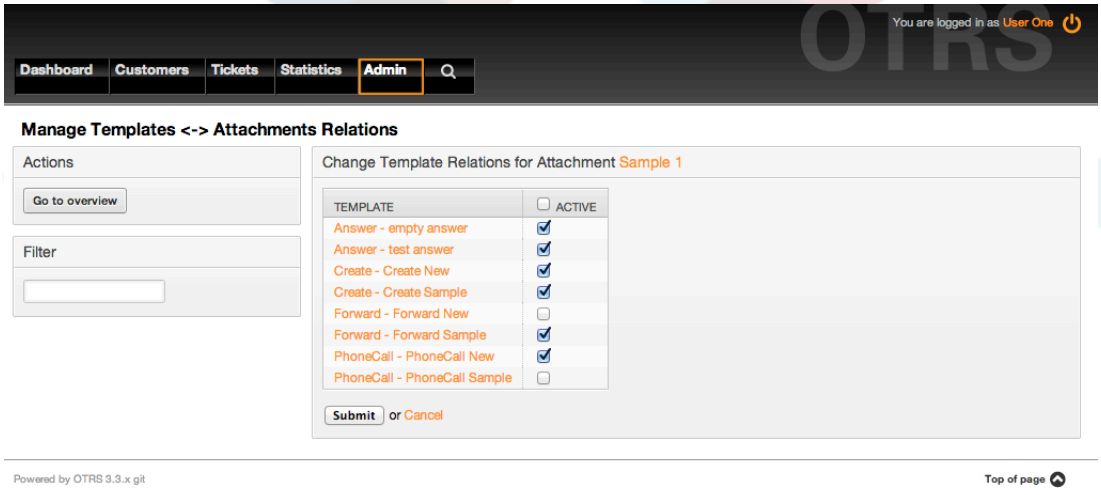


図:添付用テンプレート関連の変更

1.5.4. テンプレート

チケット処理を促進し、かつ返答の外観を標準化するために、OTRSにテンプレートを定義することができます。テンプレートは1つ以上のキューにリンク、および逆リンクも可能です。

OTRSの異なる部分で使用される、異なる種類およびそれらのテンプレートがありますそれらはそれ自身の目的を持っていて、下記は可能なテンプレート・タイプのリストです。:

- 返答：Answer: チケット・応答として使用されるか返答します。
- 作成：Create: 新規電話かEメール・チケットの中で使用されます。
- 転送：Forward: 誰か他の人へ記事を転送するために使用されます。
- 電話：PhoneCall: 電話の着信および発信画面で使用されます。

電話の受発信画面で使うためには、返答テンプレートは二つの方法でアクセス可能です。ひとつは、記事メニューでのチケット・ズーム画面から。もうひとつ、より素早い方法なら、状態ビューやチケット・ビューのようなチケット概要の大画面から。新規のOTRS導入に関しては、テンプレート「空の答え（empty answer）」がすべてのキューに対してデフォルトでセットされます。

転送テンプレートがキューに割り当てられ加えられるとすぐに、チケット・ズーム中の「転送」ボタン（通常は空のテキスト転送画面に導かれます）は選択コントロールに変化します。ひとつのテンプレートを選択することで「選択」は、転送テンプレートに情報を与えます。転送画面は、テンプレートテキストと添付で事前に情報を与えられ表示されます。（返答テンプレートでの「返答選択ボックス」と同様です。）

タイプ「作成」と「電話」のテンプレートを作成することは、
　それぞれの画面の中の「テキスト・テンプレート」選択ボックス」を見えるようにします。リストにテンプレートを選択することは、「テキスト」と「添付」フィールド（テンプレートで利用可能ならば）にデータを追加することになるでしょう。テキストまたは添付で何からの以前の変更が、テンプレートを選択することで上書きされることに気づいて下さい。

管理者ページの「テンプレート」をクリックすれば、テンプレート管理画面が開きます。(下記の図を参照)。

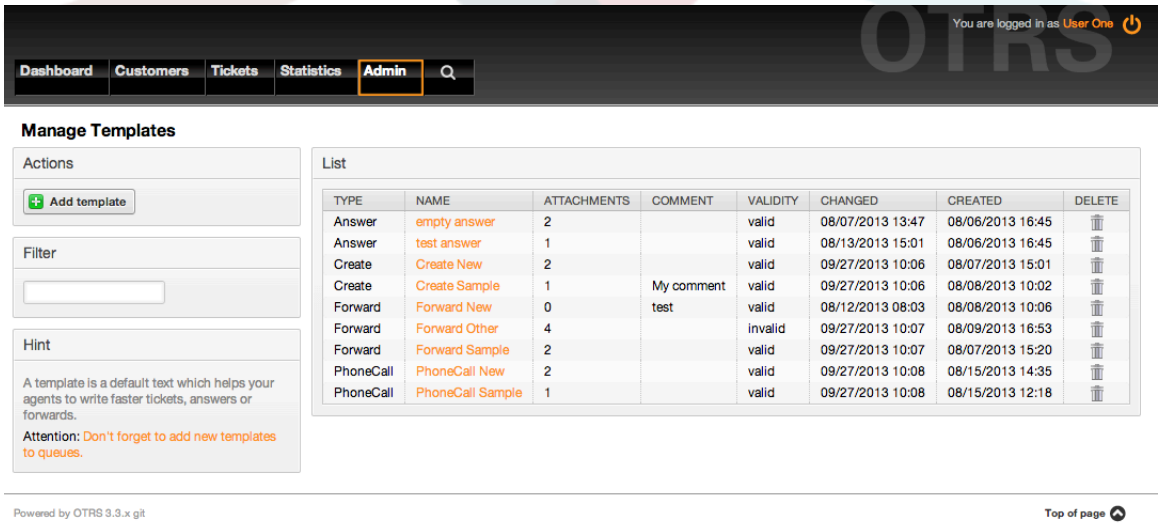


図:テンプレート管理

新規テンプレート(「テンプレート追加」ボタン上のクリック)を作成するためには、必要なデータ(確実に適切なテンプレート・タイプを選択すること)を提供して、送信下さい(下記の図を参照)。

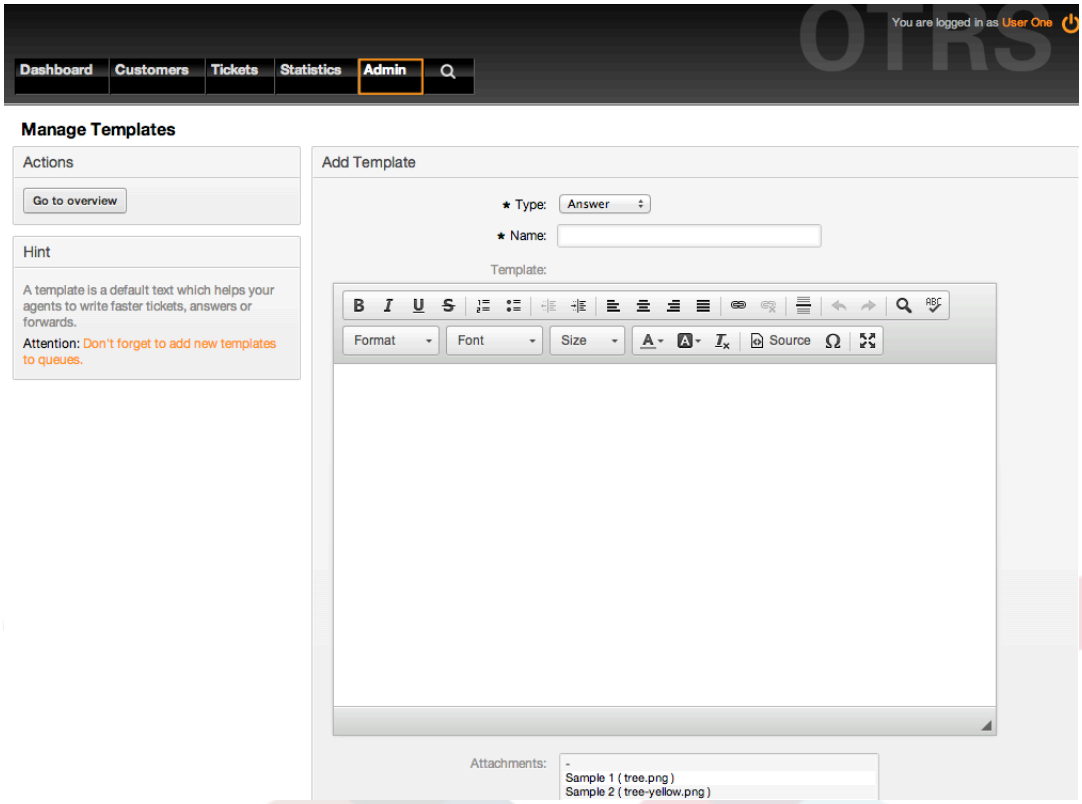


図:テンプレートの追加

テンプレートを1つ以上のキューに追加/削除するためには、管理者ページの「テンプレート<->キュー」リンクをクリックしてください(下記の図を参照)。さらに、特定の実体に関する情報を得るためにフィルタを使用することができます。

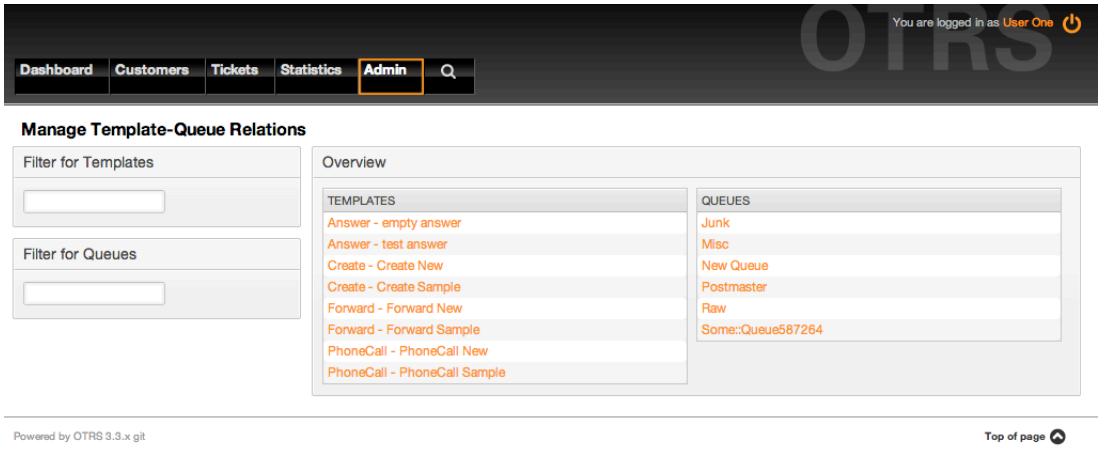


図:テンプレート・キュー関係管理

キューに、および逆に (and vice versa) 利用可能になる、異なるテンプレートを定義するためには、対応するテンプレートかキューをクリックしてください(図5.32および5.33以下にそれぞれ参照)。

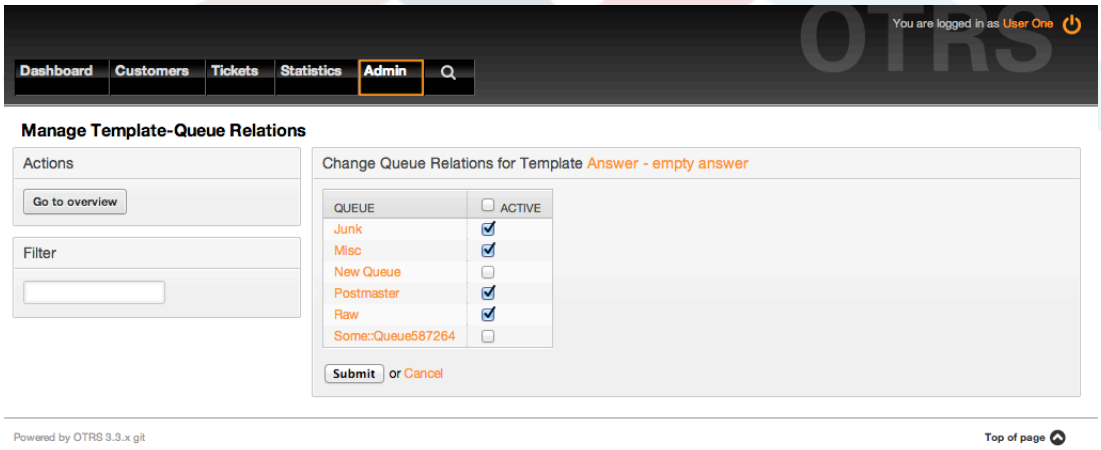


図:テンプレートのためのキュー関係の変更

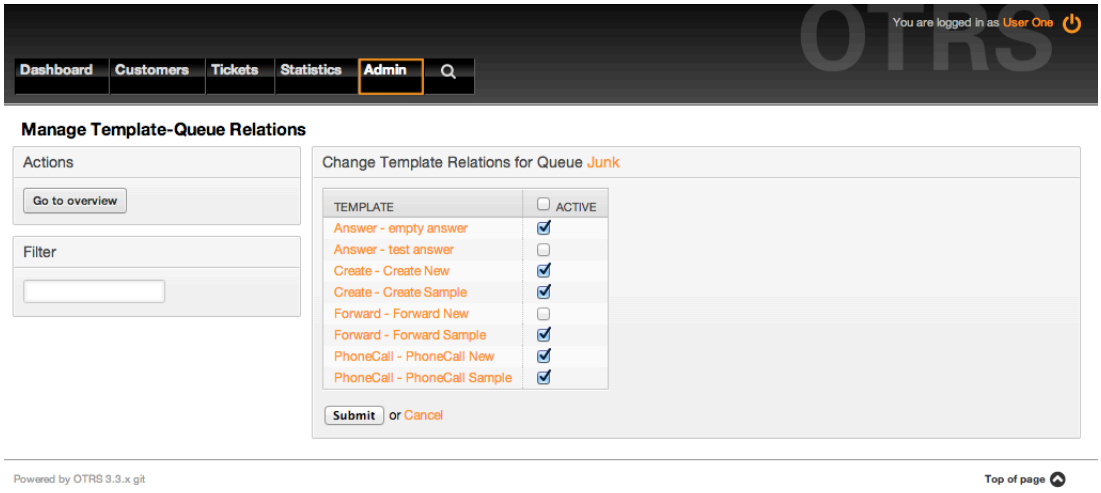


図:キューのためのテンプレート関係の変更

テンプレートを選択するとき追加の情報がテンプレート・テキストに追加できます。これはテンプレートのタイプに依存します：

しかしながら、「電話」とテンプレートの作成はテンプレート・テキストにいかなるコンテンツも追加しません。新規Eメールチケット画面はキューを追加し、そのキューは解決Eメール本文に署名を割り当てます。（この画面は署名を視覚化するための別のボックスを持ちます。）

返答テンプレートテキストは、選択された時、キューに関連づけられて挨拶文を含みます。
それからテンプレートはのテキスト、それから引用されたチケットのテキスト、そして最後はキューに関連付けられた署名も含みます。

転送テンプレートは返答テンプレートと同様ですが、それらは挨拶文部分を含みません。

1.6. 自動応答

OTRSは、特定のイベントが発生に応じて顧客へ自動応答を送信することを可能にします。それらのイベントとは、特定のキュー中のチケットの生成、チケットに関するフォローアップメッセージの受取、チケットの完了か拒否、等です。そのような応答を管理するためには、管理者ページ上でリンク「自動応答」をクリックしてください(下記の図を参照)。

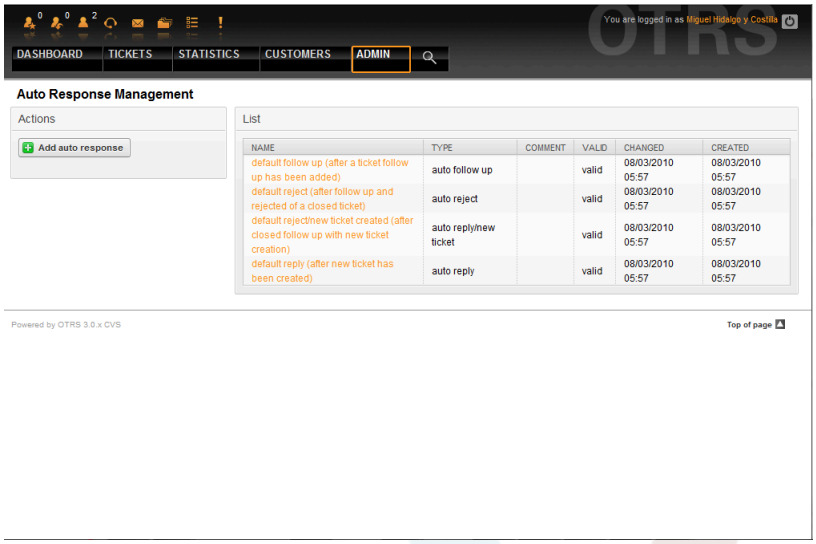
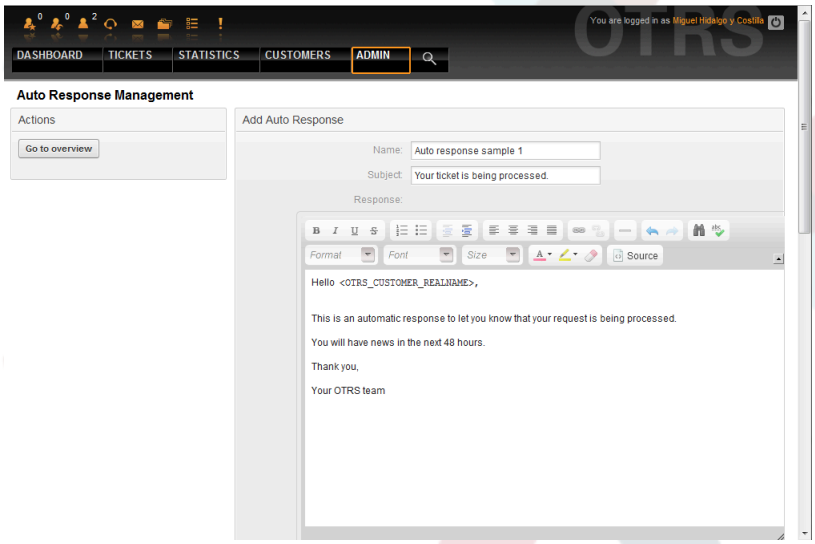


図:自動応答管理

自動応答を作成するために、「自動応答の追加」をボタンをクリックし、必要とされるデータを提供し、送信して下さい(下記の図を参照)。



図：自動応答の追加

署名と挨拶文でそうであったように、自動応答の件名およびテキストは、変数によって生成することができます。例えば、自動応答の本文に可変<OTRS_CUSTOMER_EMAIL[5]>を挿入すれば、顧客メール・テキストの最初の5行が自動応答に挿入されるでしょう。図示された画面の一番下にて使用することができる有効な変数に関するより多くの詳細を見つけるでしょう。

すべての自動返答ではそれをきっかけとすべきイベントを指定することができます。デフォルトのインストレーションの後で利用可能なシステム・イベントはテーブル5-3に述べられています。

表4.4 自動応答用イベント

Name	摘要
自動応答	あるキュー中のチケットの自動返答生成
自動応答/新規チケット	既に完了チケットへの返信。例えば顧客がそのチケットへ返信した場合
自動フォローアップ	チケットに対するフォローアップの受付

Name	摘要
自動拒否	システムによって行われたチケットの自動拒否
自動削除	システムによって行われたチケットの削除

注記

他のOTRSの実体と同様に、自動応答も単に非活性化されるだけであり、削除はできません。（有効オプションを無効または無効-暫定に設定することによって）

キューに対する自動応答を加えるためには、管理者ページ上で「自動応答<-> キュー」リンクを使用してください(下記の図を参照)。全てのシステム・イベントは全てのキューのためにリストされます。また、同じイベントを備えた自動応答はリストボックスによって選択するか、削除することができます。

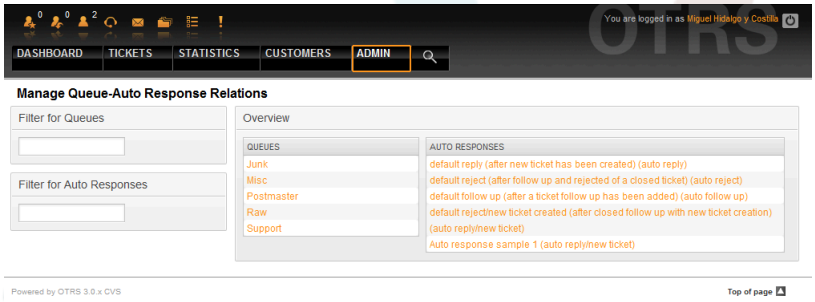


図:キュー自動応答関連の管理

キューに利用可能になる異なる自動応答を定義するためには、対応するキュー名をクリックしてください(下記の図を参照)。さらに、既存の自動応答を編集することは可能です。それには「応答」をクリックして、「新規自動応答を編集」として同じやり方で編集してください。

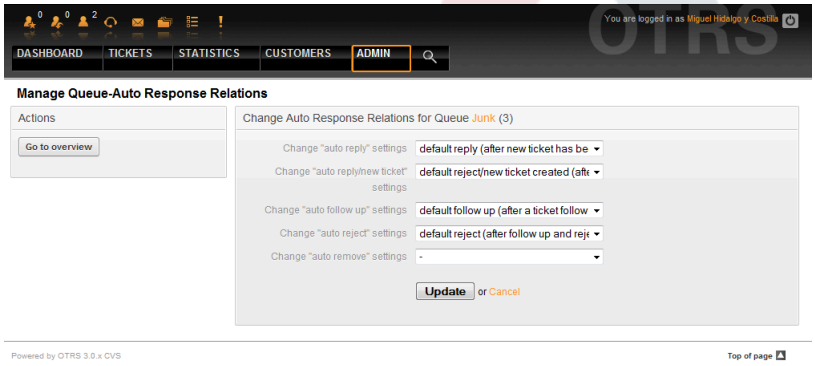


図:キュー用の自動応答関連の変更

1.7. Eメール・アドレス

OTRSがEメールを送ることを可能にするために、システムによって使用されるために有効なEメール・アドレスを必要とします。多くのサポート・インストレーションはひとつ以上の使用する必要があるので、OTRSは複数のので稼働できます。キューは多くのEメール・アドレスにリンクすることができ、その逆も可です。キューが作成されると、キューから送信メッセージとして送信するEメール・アドレスは、セットされます。システムの全てのEメール・アドレスを管理するために、管理者ページから「Eメール・アドレス」リンクを使用してください(下記の図を参照)。

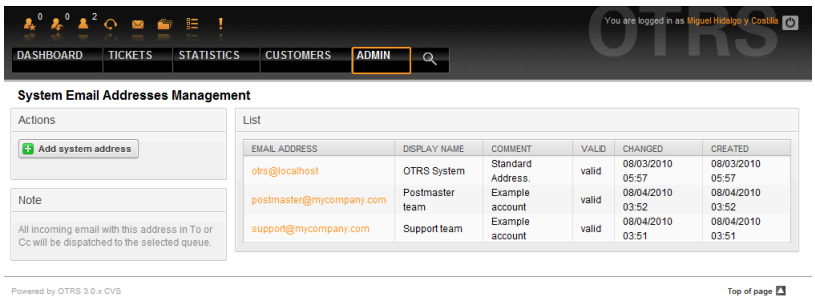


図:システム・*E*メール・アドレス管理

新規Eメールアドレス(下記の図を参照)を作れば、それとリンクされるキューかサブキューを選択することができます。このリンクでシステムは受信メッセージをソートすることが可能となります。正しいキューの中で　メールのTo;フィールドアドレスによりソートされます。

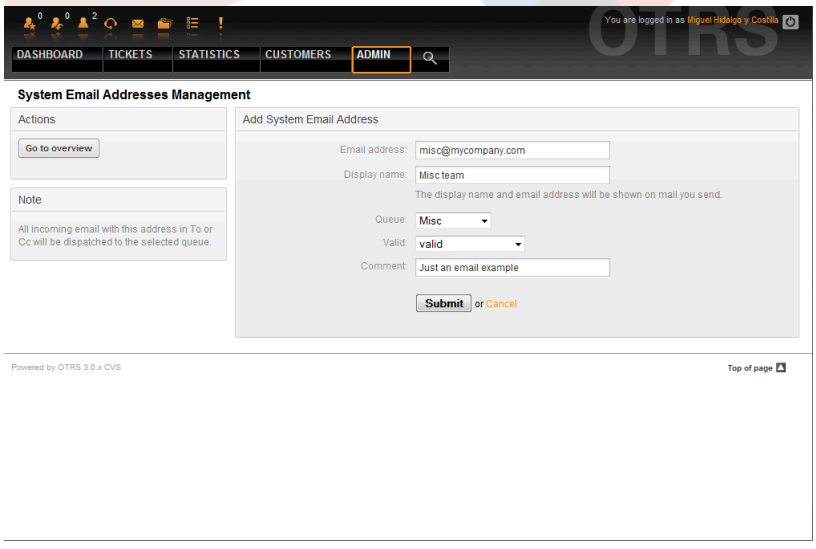


図:システム・メール・アドレスの追加

注記

他のOTRSの実体と同様に、Eメール・アドレスも単に非活性化されるだけであり、削除はできません。（有効オプションを無効または無効-暫定に設定することによって）

1.8. 通知

OTRSは、あるイベントの発生に基づいて担当者や顧客へ通知することが可能です。担当者は、選択（選好）のリンク経由で自身への通知用のシステム・イベントを設定できます。

管理者ページの「担当者通知」リンクによって、システムの通知を管理することができます(下記の図を参照)。ある通知だけをリストするためにフィルタを使用することができます。

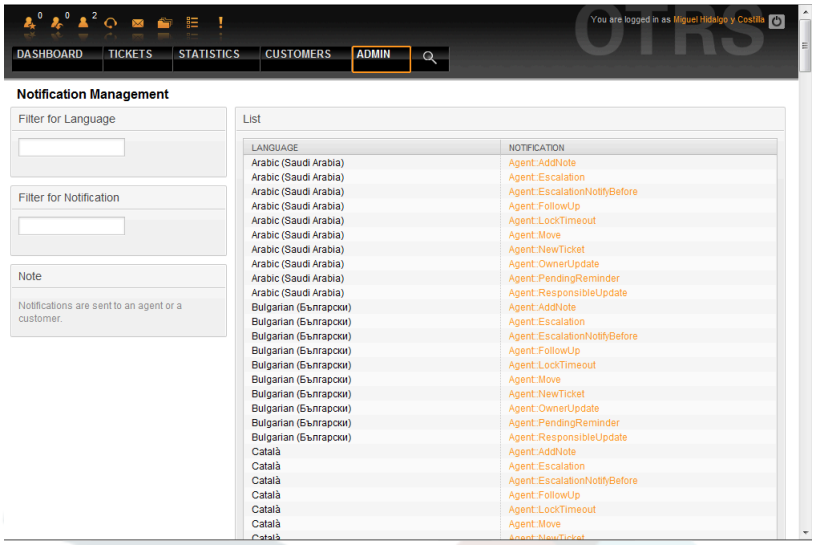
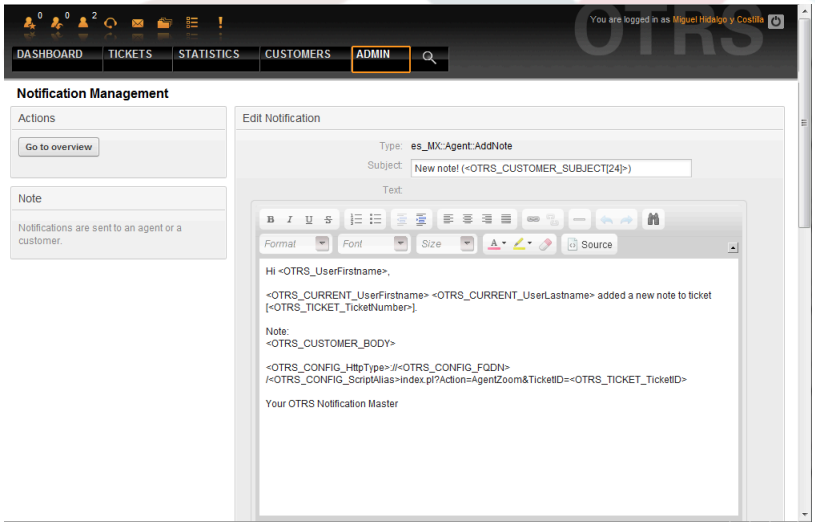


図:通知管理

件名および通知テキストをカスタマイズすることができます。リストから変更したい通知をクリックしてください(図を参照)。そうすれば、内容が編集用にロードされます。利用可能な言語の各々の同じ名前を持った通知があることをご留意ください。



通知のカスタマイズ

署名と挨拶文でそうであるように、特殊変数の使用により、通知の内容を動的に作成することは可能です。図の中で示される画面最下部の変数のリストを見つけることができます。

さらに、イベントに基づいた通知を作成することは可能です。誰がいつあなたにそのような通知を送信するか、を詳細に指定することができます、以下のような様々のパラメタから選択することができます: 受信グループ, 担当者, 役割, Eメールアドレス, 通知契機イベントタイプ, チケットタイプ, 状態, 優先度, キュー, ロック, サービス, SLA, 等。

すべてのイベントに基づいた通知のリストを見るためには、管理者ページ上でリンク「通知(イベント)」をクリックしてください(図を参照)。

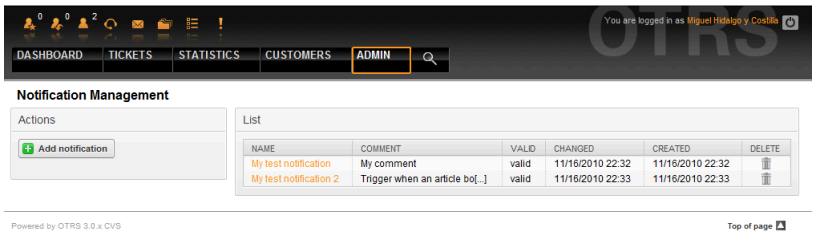


図:通知管理に基づくイベント

図の中で示されるように、追加ボタンをクリックすることにより新規通知を作成することができます(図を参照)。

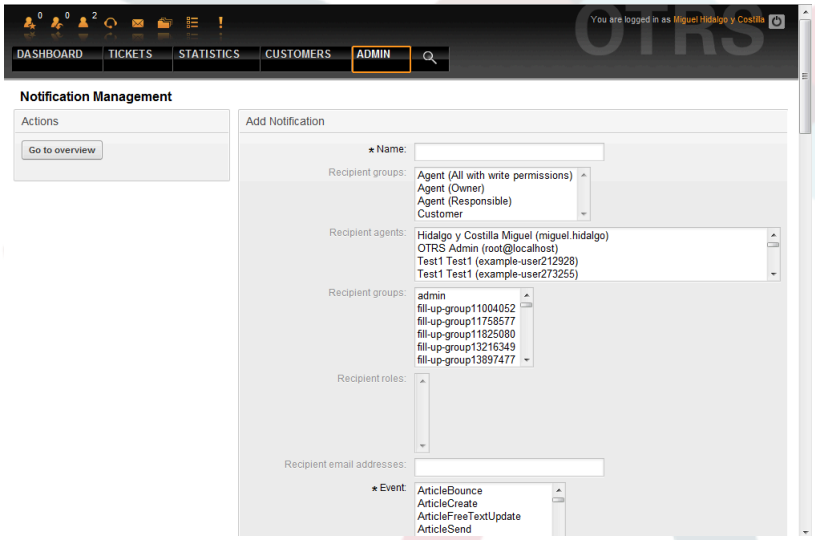


図: 通知管理に基づくイベントの登録

図の中で示される画面の一番下にリストされた特殊変数の使用により、通知に基づいたイベント内容をダイナミックに作成することができることにご留意ください。

1.9. SMIME

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

管理者ページの「S/MIME証明書」リンクによってS/MIME証明書を管理することができます(下記の図を参照)。証明書の追加及び削除、またSMIMEデータの検索もできます。

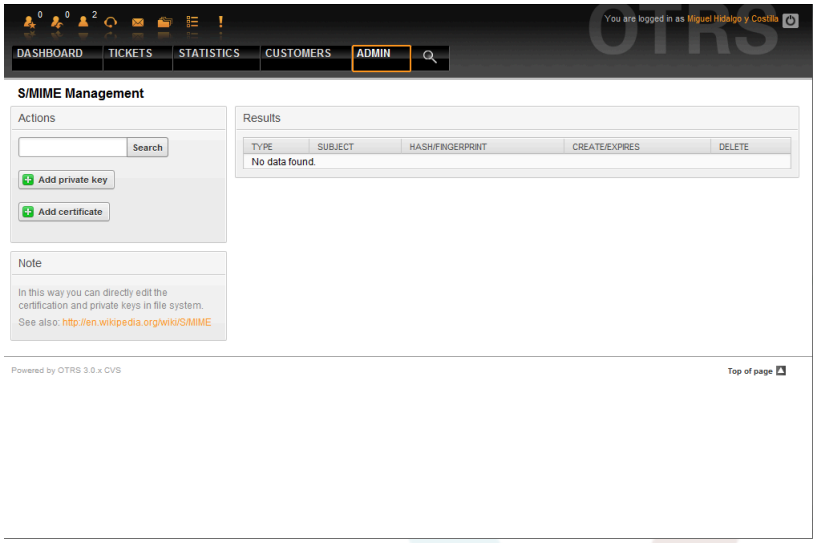


図: *S/MIME* 管理

1.10. PGP鍵

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.

管理者ページの「PGP鍵」リンクによって、OTRS(例えば地方のOTRSユーザ、ウェブサーバ・ユーザ)(下記の図を参照)を備えたPGP鍵に使用されるものとするユーザのkey ringを管理することは可能です。キーと署名を加えること、削除することは可能です。また、key ring中のすべてのデータを捜すことができます。

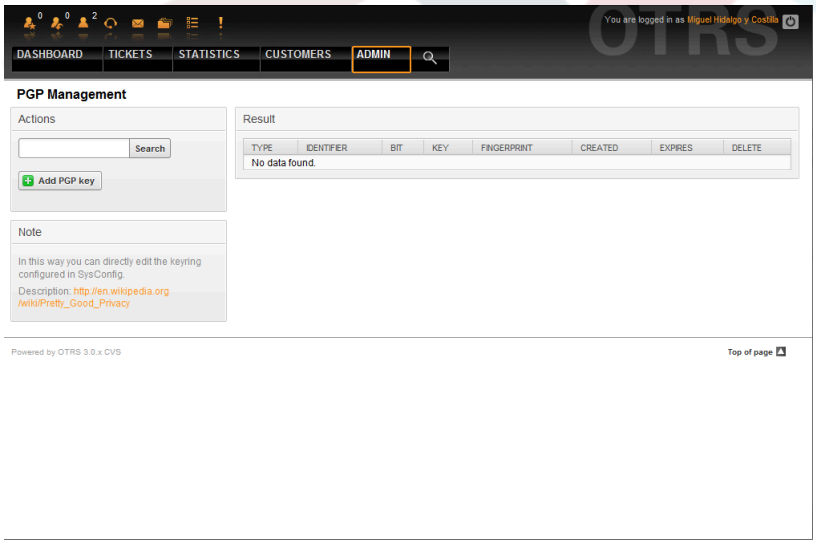


図: *PGP*鍵 管理

1.11. 状態

管理者ページの「状態」リンクによって、システムの中で使用したい異なるチケット状態を管理することができます(下記の図を参照)。

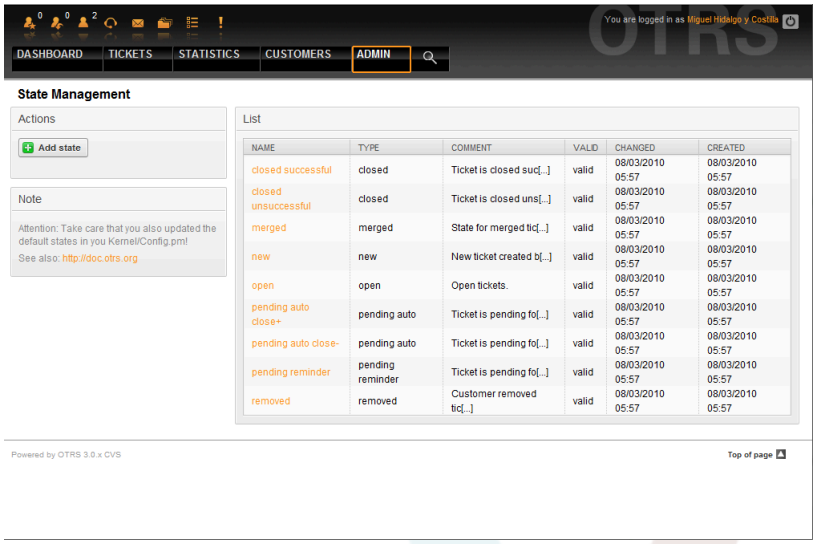


図:状態管理

デフォルトの開始後、定義される「状態」がいくつかあります。

- 完了（成功）
- 完了（不成功）
- 結合済
- 「新規」（new）
- 対応中（open）
- 保留（自動完了＋）
- 保留（自動完了-）
- 保留（期限付）
- 削除済

全ての状態はタイプにリンクされます。それは新規状態が作成されならば指定されねばなりません。デフォルトの状態タイプは次の通りです。

- 完了済
- 結合済
- 「新規」（new）
- 対応中（open）
- 保留（自動）
- 保留（期限付）
- 削除済

1.12. SysConfig

SysConfigのリンクは、多くのOTRSコンフィグレーション・オプションがメンテナンスされるセクションへと辿り着きます。

管理者ページのSysConfigリンクは、システム構成用のグラフィカル・インターフェースをロードする(下記の図を参照)。　現在の全ての設定を 1 ファイルにバックアップするように、システム用の

自分の構成ファイルをアップロードすることができます。OTRSフレームワークおよびインストールされたアプリケーションの構成パラメタはほとんどすべて、このインターフェースを通って見て変更することができます。全ての構成パラメタがグループおよびサブグループへソートされるので、既存のパラメタの莫大な数によってすばやく閲覧することは可能です。さらに、構成パラメタのすべてによって全文検索を行なうことも可能です。

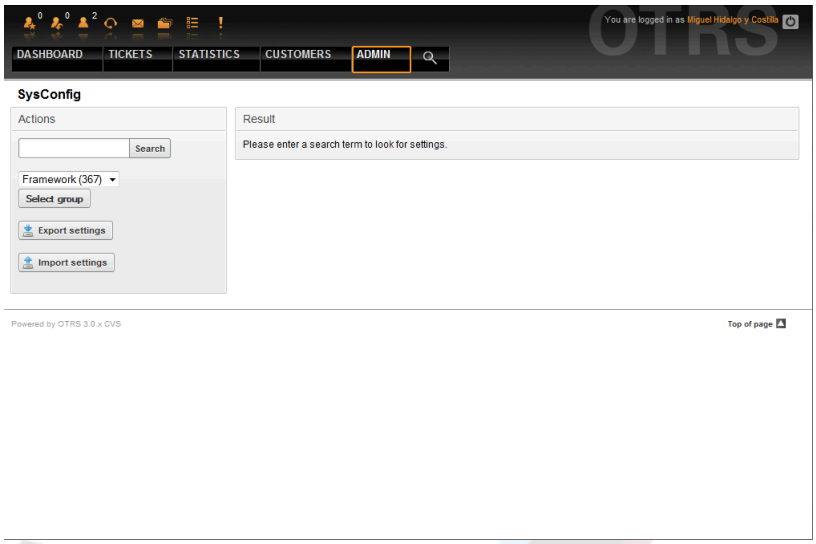


図:システム構成*(SysConfig)*用のグラフィカル・インターフェース

システム構成用グラフィカル・インターフェイスに関してこの章　"ウェブ・インターフェイスからのシステム構成"でより詳細に記述されます。

1.13. Eメールアカウントの使用

チケット・システムへ新規Eメールを移送する可能性いくつかあります。ひとつはローカルMTAとシステムに直接メールを繋ぐotrs.PostMaster.pl　scriptです。もうひとつの可能性は、ウェブ・インターフェイスをと通して管理されるメールアカウントを使用することです。管理ページにある"ポストマスター・メールアカウント"リンクは、メールアカウント(下図参照)管理コンソールをロードします。OTRSがサポートするメール・プロトコル：POP3, POP3S, IMAPとIMAPSです。

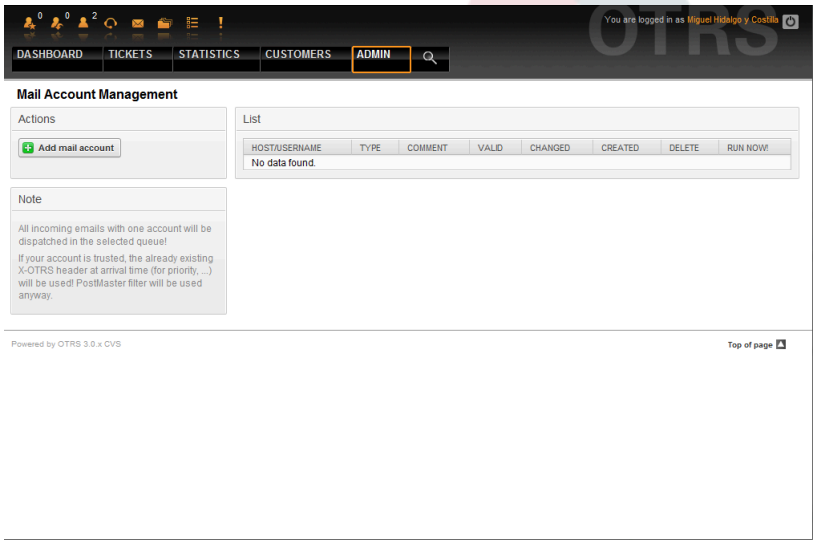


図: *E*メールアカウント管理

より詳細については、PostMasterメールアカウント に関するセクションを参照してください。

1.14. 受信Eメール・メッセージのフィルタリング

OTRSは、受信Eメール・メッセージのフィルタリング能力を持っています。例えば、指定されたキューに特定のEメールを自動的に入れるか、いくつかのメールのために特定の状態かチケット・タイプをセットすることが可能です。フィルタはすべての受信メールに適応します。管理者ページ上でリンク「PostMasterフィルタ」によってフィルタを管理することができます(下記の図を参照)。

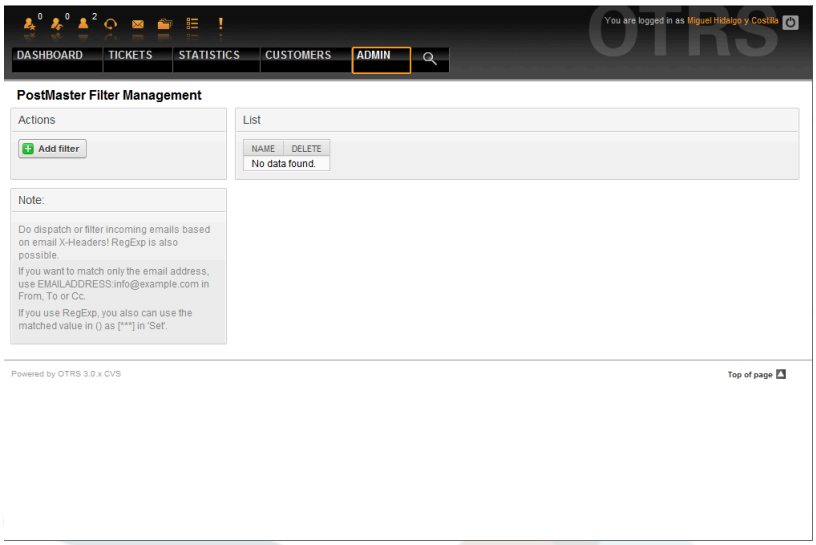


図:ポストマスター・フィルタ管理

フィルタは、Eメール上で実行される定義されたアクションのために、順番に合致しなければならない1つ以上の基準から成ります。フィルター基準は、本文にヘッダーあるいはEメールの本文のために定義できます。(例えば送信者アドレスや本文中の文字列などのような特定のヘッダー入力の検索)正規表現でさえ拡張パターン一致のために使用することができます。フィルタが一致する場合、GUIの中でX-OTRSヘッダーを使用して、フィールドをセットすることができます。OTRSでチケットかフォローアップ・メッセージを作成する時、これらの値が適用されるでしょう。テーブル5-4は異なるX-OTRSヘッダーおよびそれらの意味をリストします。

さらに、フォローアップEメールに対して値をセットするために X-OTRS-FollowUp-*ヘッダーを使用することができます。

表4.5 異なるX-OTRSヘッダーの機能

Name	可能な値	摘要
X-OTRS-優先度:	1 極低い, 2 低い, 3 普通, 4 高い, 5 極高い	チケットの優先度を設定します。
X-OTRS-キュー:	システム内でのキュー名	チケットがソートされる場合、キューをセットします。もしX-OTRSヘッダーの中でセットされれば、特定のキューへチケットをソートしようとする他のすべてのフィルタ規則が無視されます。サブキューを使用する場合は、「Parent::Sub」としてそれを指定してください。
X-OTRS-ロック:	ロック、アンロック	チケットのロック状態を設定します
X-OTRS-無視:	「はい」 あるいは 「真」	このX-OTRSヘッダーが「はい」にセットされれば、受信

Name	可能な値	摘要
		メッセージは完全に無視され、システムに配達されないでしょう。
X-OTRS-状態:	新規、対応中、完了（成功）、完了（不成功）...	チケットの次の状態を設定します
X-OTRS-状態-保留時間:	例えば 2010-11-20 00:00:00	チケットの保留時間を設定します。（X-OTRS-Stateによって保留状態を送信もして下さい）電子メールの到着時刻に基づいて、「2010-11-20 00:00:00」のような絶対日付あるいは相対日付を指定することができます。フォーム「+ $Number $Unit」を使用してください。ここで、「$Unit」は、's'の(秒)、「m」(分)、「h」(時間)あるいは「d」(日)です。1ユニットのみ指定することができます。有効な設定例:「+50s」(50秒でペンディング)、「+30m」(30分)、「+12d」(12日)。「+1d 12h」のようなセッティングができないことに注意してください。その代わりに「+36h」を指定できます。
X-OTRS-タイプ:	デフォルト (あなたの実施したセットアップに依存します)	チケットタイプの設定 (Ticket::Type が活性化されていれば)。
X-OTRS-サービス:	（セットアップにもよりますが）	チケットのサービスを設定します。(Ticket::Service がアクティブならば)。サブキューを使用する場合は、"Parent::Sub" としてそれを指定してください。
X-OTRS-SLA:	（セットアップにもよりますが）	チケットのSLAを設定します。(Ticket::Service support がアクティブならば)。
X-OTRS-顧客ユーザ:	CustomerUser（顧客ユーザ）	チケット用顧客ユーザーを設定します。
X-OTRS-顧客番号:	CustomerNo（顧客番号）	このチケットの顧客IDを設定します。
X-OTRS-送信者タイプ：	担当者、システム、顧客	このチケット送信者のタイプを設定します。
X-OTRS-記事タイプ:	外部Eメール、内部Eメール、外部Eメール通知、内部Eメール通知、電話、ファックス、SMS, ウェブリクエスト、内部ノート、外部ノート、ノートレポート。	受信チケットの記事タイプを設定します。
X-OTRS-DynamicField-<DynamicFieldName>:	動的フィールドの構成に依存します（テキスト：ノートブック、日付：2010-11-20 00:00:00, Integer: 1)	摘要:動的フィールド <DynamicFieldName> でチケットの追加情報値を保存します。

Name	可能な値	摘要
X-OTRS-ループ:	真	このX-OTRSヘッダーが設定されたら、メッセージ送信者へ自動返信は配信されません。（メール・ループ保護のため）

すべてのフィルタ・ルールに対して名前を指定するべきです。フィルター基準はセクション「フィルタ条件」の中で指定することができます。「ヘッダー1」、「ヘッダー2」あるいは、あなたが検索したいメッセージの部分等をリストボックスから選んでください。そしてあなたがフィルターしたい値を右側で指定して下さい。セクション「セット電子メール・ヘッダー」では、フィルタ規則が一致する場合、起きるアクションを選ぶことができます。X-OTRSヘッダーを選択し、かつ関連する値(下記の図を参照)をセットするために、「ヘッダー1」、「ヘッダー2」などを選択することができます。フィルタ規則はアルファベティカルに評価され、すべて実行されます。但し、「Stop after match」セッティングが規則のうちの1つで「はい」にセットされている場合を除きます。(この場合、残りのフィルタの評価は取り消されます。)

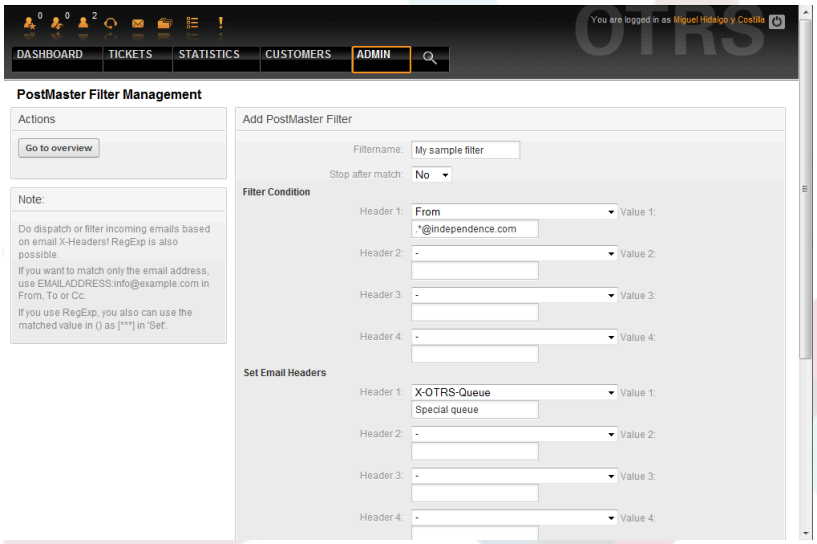


図:PostMasterフィルタの追加

例4.1 特定のキューの中へのスパム・メールのソート

OTRSは、有用なフィルタ規則によってスパムと判定されたEメールを自動的に移動します。SpamAssassinのようなスパム検出ツールによって「ジャンク」キューへ移動します。SpamAssassinは「X-Spam-Flag」ヘッダーを全てのチェックしたメールに追加します。メールがスパムとして判定された場合、ヘッダーは「はい」にセットされます。したがって、フィルター条件は「X-Spam-Flag」が「Yes」になります。この基準でフィルタ規則を作成するために、例えば「spam-mails」として名前を挿入することができます。「フィルタ条件」のためのセクションでは、リストボックスあｋら「Header 1」に対して「X-Spam-Flag」を選んでください。このヘッダーに見合う値として「はい」を挿入してください。今、フィルター条件は指定されます。スパム・メールがすべて「ジャンク」キューに入れられることを確認し、「Set Email Headers」のためのセクションで「Header 1」のために入力した「X-OTRS-Queue:」を選択して下さい。このヘッダーの値として「ジャンク」を指定してください。最後に、システムでの新規メッセージを活性化する、新規フィルタ規則を追加して下さい。

追加モジュール(受信メッセージのフィルタリングをより明確にするために使用することができる)があります。より大きく、より複雑なシステムに対処する場合、これらのモジュールは有用かもしれません。

1.15. GenericAgentでの自動仕事の実行

GenericAgent (一般担当者) はタスクを自動的に実行するツールです。「一般担当者」は例えば、チケットを完了したり、移動したり、エスカレートされたチケットに対して通知を送信したりできます。

管理者ページ上でリンクGenericAgen (一般担当者) をクリックしてください(下記の図を参照)。システムでのすべての自動ジョブを備えたテーブルが表示されます。その後、これらのジョブは編集、手動で動作、または完全に削除することができます。

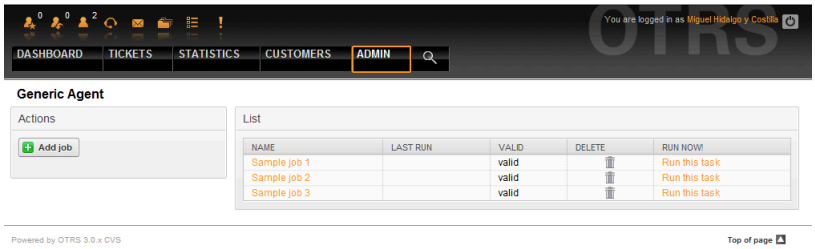


図: *GenericAgentのためのジョブ・リスト*

新規ジョブを作成するために「ジョブ追加」ボタンをクリックしてください。最初に名前を提供する必要があります。その後、ジョブがどのように実行されるか特定することができます:指定時刻で自動 (cronjobのようにこのモードはチケット・フィルターによってすべてのチケットで作動します。) あるいは、チケット・イベンで自動 (チケット・フィルターにマッチしていれば単一のチケットが変更された直後に) 。

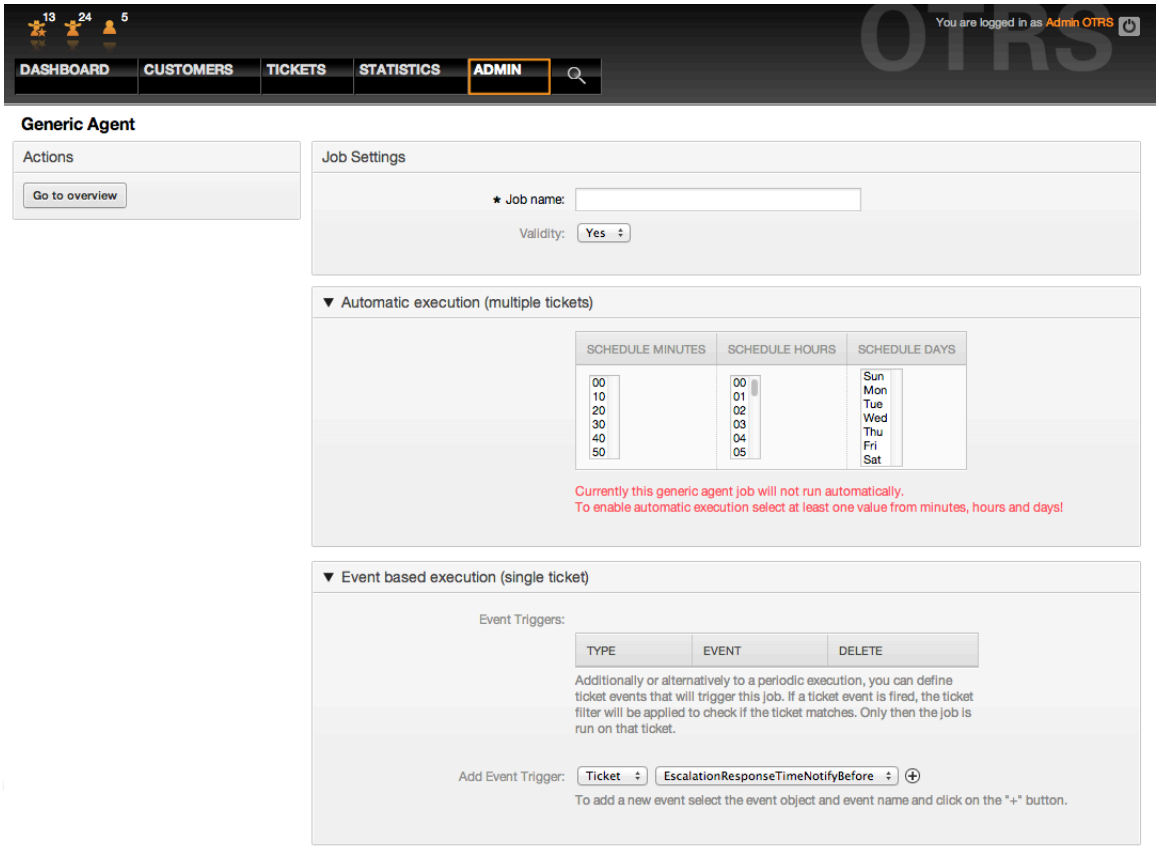


図: *GenericAgentのためのジョブ作成*

すべてのジョブについては、チケット・フィルタを指定することができるが、例えば、ある特定のキュー中ではチケット上でのみ作動します。フィルター条件はすべて、チケット上で実行されるジョブのために満たされなければならない。

すべてのジョブについては、チケット・フィルタを指定することができるが、例えば、ある特定のキュー中ではチケット上でのみ作動します。フィルター条件はすべて、チケット上で実行されるジョブのために満たされなければなりません。最後に、チケットは新規キューか状態のような様々なチケット・フィールドの設定により修正することができます。チケットにノートを付けるか、カスタマイズされたモジュールを実行することは可能です。さらに、データベースからチケットを削除するオプションがあります。これはシステムからの時代遅れのデータあるいは無効のデータを除去するのに有用になりえます。

警告

チケットを削除する機能を使う場合、すべての影響を受けるチケットやそれらの添付ファイルはデータベースから取り除かれ復元することはできません。

ジョブを編集した後に、OTRSは概要画面に戻ります。そこで、どんなジョブも手動で実行できる可能性があります。ジョブを実行することを選ぶならば、ジョブが実際に走っている時に全てのチケットが影響を受けることをまず最初に見ることでしょう。このリストは、あなたが意図通りにジョブが働いていることを確認するのに役立ちます。このポイントでは、これらのチケットにまだ変更は行なわれていません。あなたが画面を確認する場合のみ、ジョブが実行されます。

1.16. 管理者Eメール

OTRS管理者は特定のユーザあるいはグループへメッセージを送ることができます。管理者ページの「管理者通知」リンクでは、通知されるべき担当者とグループは選択可能な画面を開きます。（下図参照）

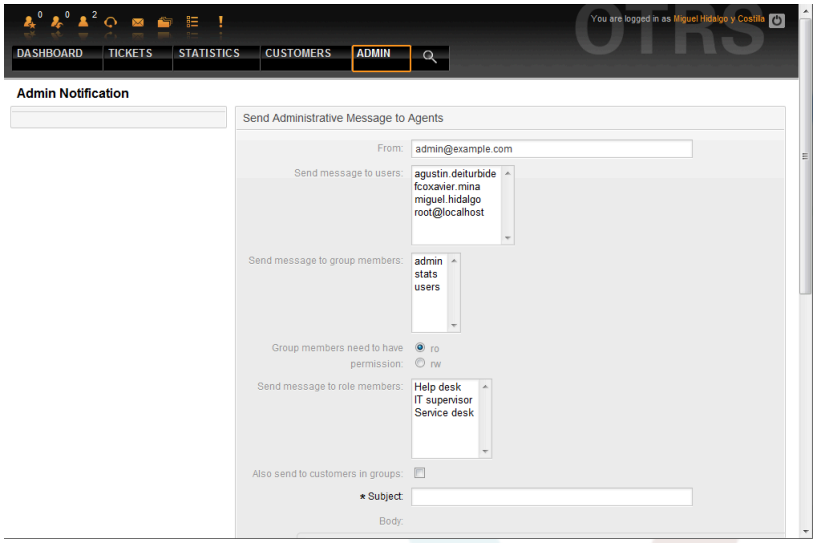


図:管理者通知

通知文の送信者、件名、本文は指定可能です。担当者、グループ、およびメッセージを受け取るべき役割を選ぶこともできます。

1.17. セッション管理

管理者エリアの「セッション管理」リンクをクリックすれば、全ログインユーザとそのセッション詳細を見ることができます(下記の図を参照)。

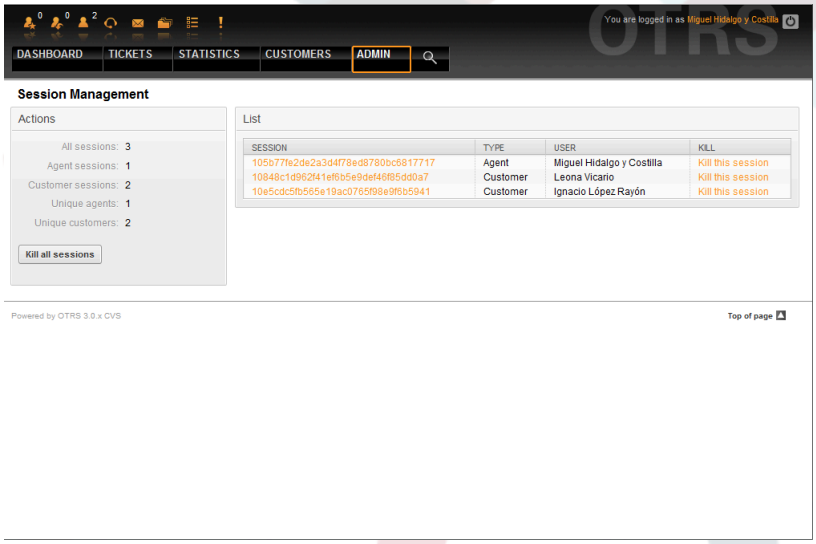


図:セッション管理

すべてのアクティブなセッションに関する統計が表示されます、例えば、ユーザはどれだけの担当者および顧客にログインされ、アクティブなセッションの数。どんな個々のセッションもリストの右側にあるこのセッションを強制終了しますリンクをクリックすることにより削除することができます。さらに、全てのセッションを強制終了しますのオプションもあり、それはシステムをオフラインで利用したければ便利です。すべてのセッションの詳細情報も利用可能です(下図参照)。

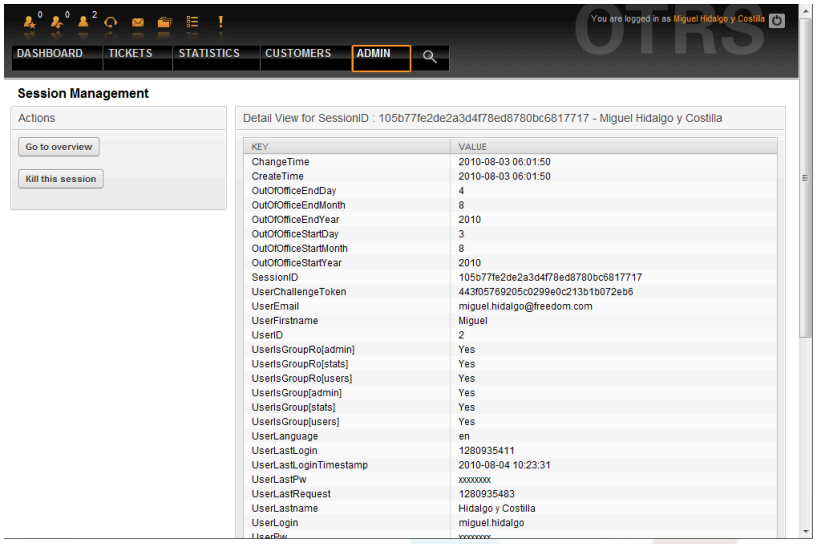


図: セッション詳細

1.18. システムログ

管理者ページの「システムログ」リンクは、システムのログ・エントリーを表示します。最新を先頭にして年代を逆順にソートされます(下記の図を参照)。

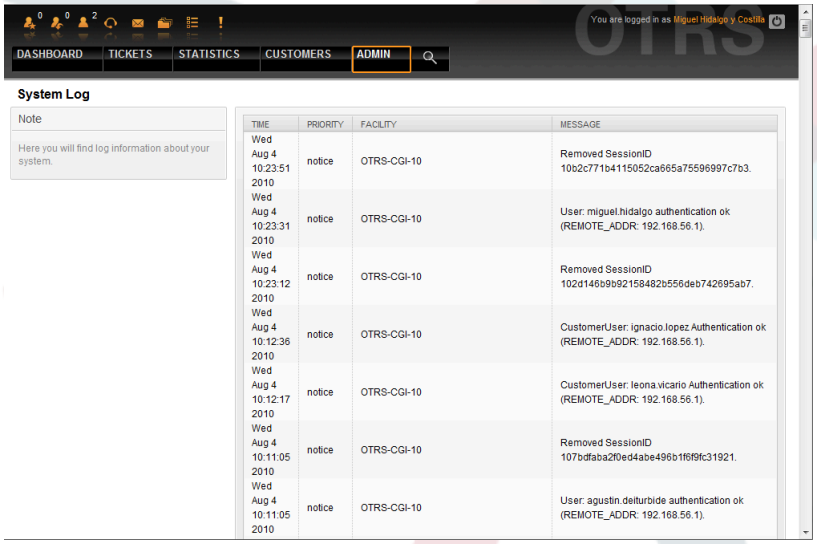


図: システムログ

ログの各ラインは、以下を含みます。タイム・スタンプ、ログ優先度、システム・コンポーネント及びログ・エントリー自体です。

注記

システムログは、Linux / Unix systems上でのみウェブインターフェース経由で利用可能です。Windows systemsではテキストエディタでファイル：[install_dir]otrs\var\log\otrs.log.を開けると、ログを見ることができます。

1.19. SQLボックス経由でのSQLクエリー

管理者ページの「SQLボックス」リンクから画面を開きます。画面では、OTRSデータベースでのテーブル・コンテンツをクエリーできます。(下記の図を参照)テーブルの内容を変更することは不可能で、「select」クエリーのみ許可されます。

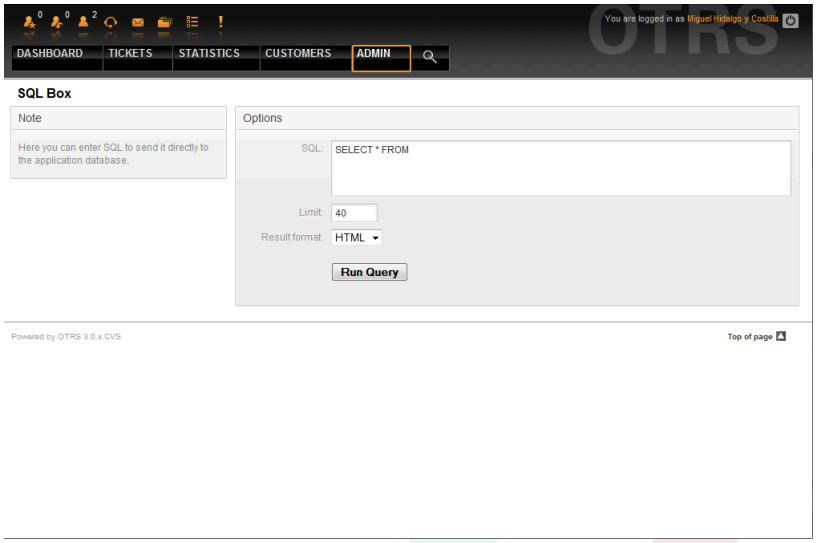


図:SQLボックス

1.20. パッケージ・マネージャー

管理者ページ上で「パッケージ・マネージャー」リンクを使用すると、OTRSの機能性を拡張するパッケージをインストールし管理することができます(下記の図を参照)。OTRSリポジトリから利用可能な拡張に関する議論は追加アプリケーションセクションを御覧ください。

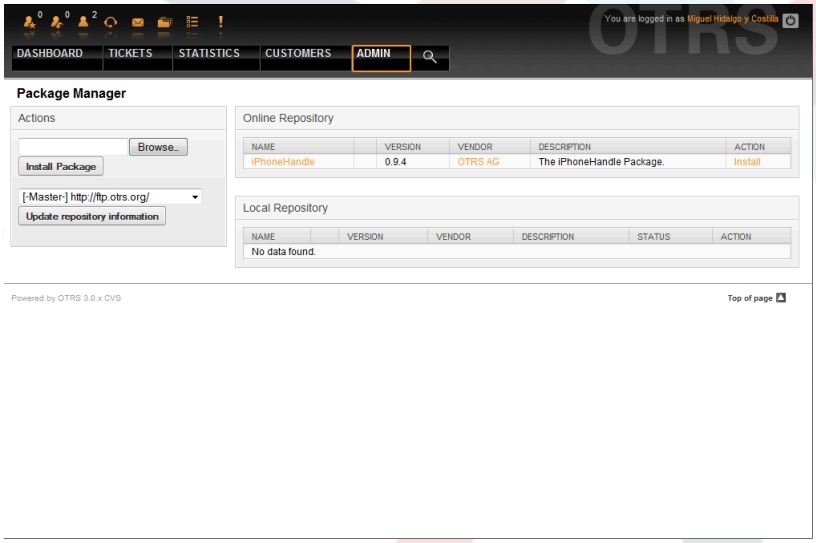


図:パッケージ・マネージャ

パッケージ・マネージャは、それらのバージョン番号と一緒に、サーバに現在インストールしたOTRS addonパッケージを示します。

オンライン・レポジトリセクションにあるレポジトリを選択し、リポジトリ情報の更新ボタンをクリックすることによって、リモート・ホストからパッケージをインストールすることができます。利用可能なパッケージが対応するテーブルに表示されます。 画面右側には利用可能なパッケージを表示します。パッケージをインストールするためにインストールをクリックします。インストレーション後、パッケージはローカル・レポジトリセクションを表示します。

インストールされたパッケージをアップグレードするために、オンライン・リポジトリにある利用可能なパッケージのリストは、ローカルにインストールされているものより高いバージョンがあるあらゆるパッケージ用のアクション・カラムでアップグレードを表示します。単にアップグレード

をクリックしてください。そうすれば、新しいパッケージ・バージョンをシステムにインストールします。

OTRSシステムがインターネットに接続されないような場合、ローカル・ディスクにダウンロードしたそれらのパッケージもインストールすることもできます。アクション・サイドバー上のブラウズをクリックし、あなたのディスク内の.opmファイルを選択します。開くを、それからパッケージをインストールをクリックします。インストレーションの完了後、パッケージはレポジトリセクションに表示されます。設置が終わった後、パッケージはのローカルのリポジトリセクションに表示されます。既にインストールされているパッケージの更新のために同じステップを使用することができます。

In special cases, you might want to configure the Package Manager, e.g., to use a proxy or to use a local repository. Just take a look at the available options in SysConfig under Framework:Core::Package.

1.21. ウェブサービス

Webサービスリンクはグラフィカル・インターフェースにつながっています。そこではウェブサービス (for the OTRS Generic Interface) が作成され維持されます。 (下図参照)

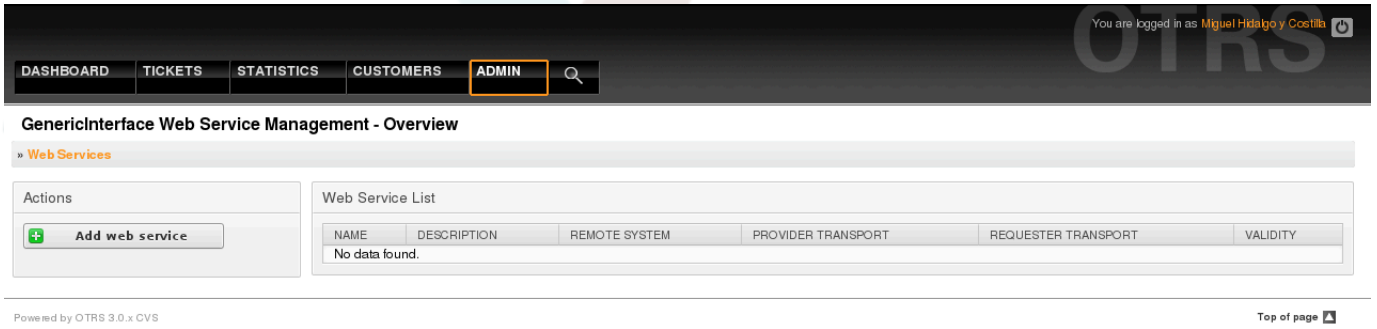


図: ウェブサービスのためのグラフィカル・インターフェース

ウェブサービス構成用のグラフィカル・インターフェースは、セクション"Webサービス・グラフィカル・インターフェース"に、より詳細に述べられています。

1.22. 動的フィールド

動的フィールドは、チケットと記事用のカスタムフィールドをセットアップし管理する場所です。 (下記参照)

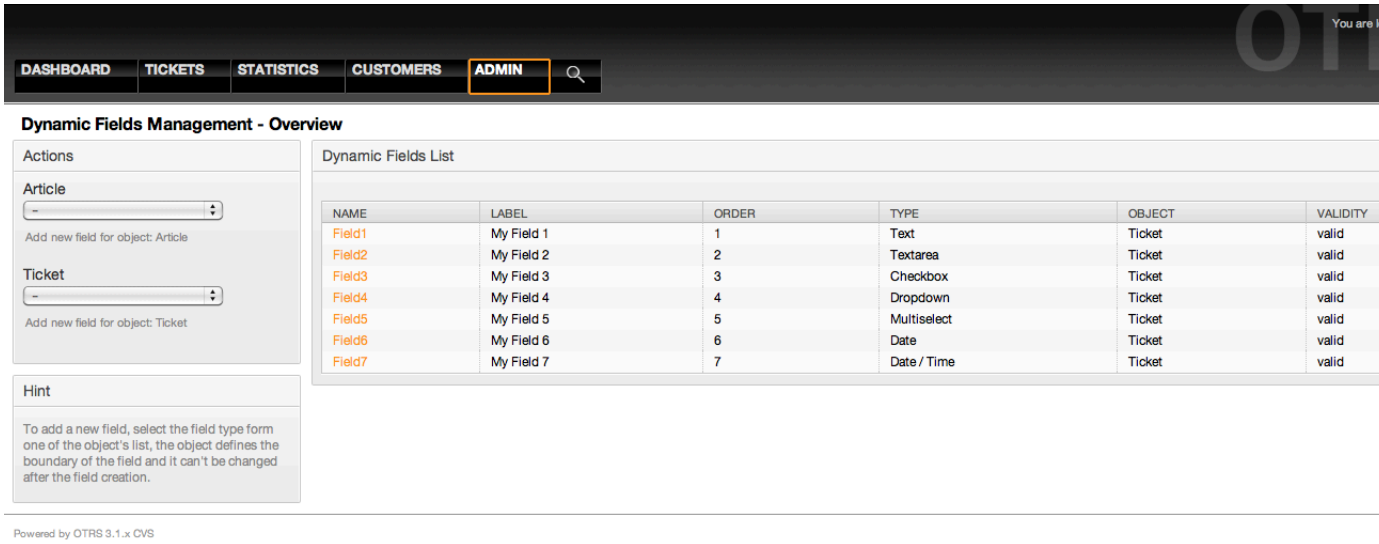


図: いくつかの動的フィールドを備えた動的フィールド概要画面

動的フィールド構成は、セクション"動的フィールド構成"に、より詳細に述べられています。

動的フィールドタイプにはそれぞれそれ自身の構成セッティングおよびしたがってそれ自身の構成画面があります。

注記

OTRSフレームワークでは、動的フィールドはデフォルトでチケットと記事にリンクのみすることができます。しかし、それらは、他のオブジェクトまで同様に延長することができます。

2. システム構成

2.1. OTRS構成ファイル

OTRS構成ファイルはすべてディレクトリKernelおよびそのサブディレクトリに格納されます。システムがアップグレードされる時、残りのファイルは変更されるので、Kernel/Config.pm以外のファイルを手動で変更する必要はありません。構成パラメータを他のファイルからKernel/Config.pmに単にコピーして、必要に応じてそれらを変更してください。このファイルがアップグレード・プロセス中に接触されることはなので手動構成は安全に実施できます。

ディレクトリ Kernel/Config/Filesでは、OTRSログイン・ページにアクセス時に、解析される他のいくつかのファイルがあります。FAQまたはファイル・マネージャーのような追加アプリケーションがインストールされている場合、それらのための構成ファイルも上記パスで見つけることができます。

OTRSウェブ・インターフェースがアクセスされる場合、Kernel/Config/Filesディレクトリ中の.xmlファイルはすべて、ABC順と設定でで解析されます。また、中央のフレームワークおよび追加アプリケーションのための設定がロードされるでしょう。後で、Kernel/Config/Files/ZZZAAuto.pm、Kernel/Config/Files/ZZZAuto.pm とKernel/Config/Files/ZZZProcessManagement.pm といったファイル(もし存在していれば)の設定は、評価されるで

しょう。これらのファイルはグラフィカル・インターフェースによってシステム構成キャッシングに使用され、手動で変更されるべきではありません。最後に、個々のセッティングおよび手動で変更された構成パラメータを含んでいるファイル Kernel/Config.pmは解析されるでしょう。この順に構成ファイルを読めば、あなたの特別な構成設定がシステムによって使用されることを確実にします。

2.2. ウェブインターフェースによるシステムの構成

OTRS　2.0以降は、中央フレームワークまたは追加でインストールされたアプリケーションのほぼすべての構成パラメタは、システム構成用グラフィカル・インターフェースで簡単に変更することができるので、新規構成ツールを実行するため、OTRS管理者としてログインして、管理者ページのSysConfigリンクをたどってください(下記の図を参照)。

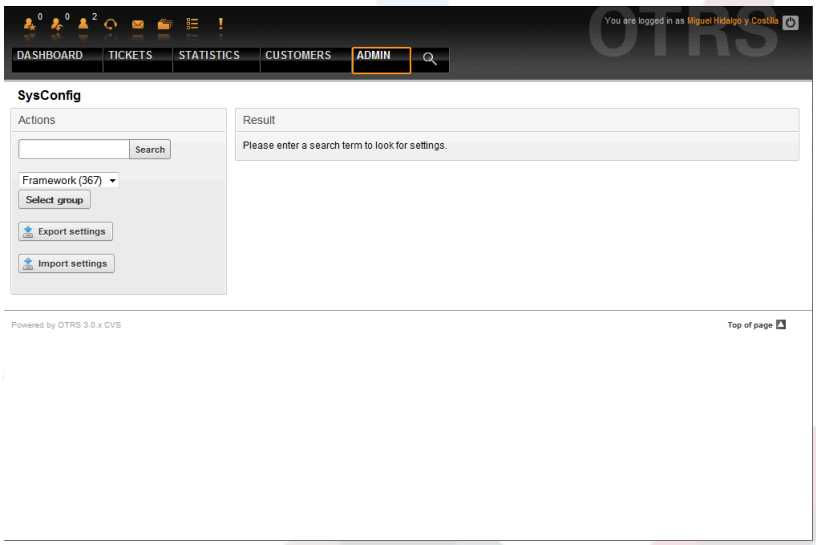


図:システム構成用のグラフィカル・インターフェース

OTRSは、現在600以上の構成パラメタを持っていて、特定のものに素早くアクセスする異なる複数の方法があります。全文検索で、構成パラメタはすべて1つ以上のキーワードでスキャンできます。全文検索は、構成パラメタの名前だけでなくパラメタの記述を通じても検索します。これによりその名前を知らなくとも要素を容易に見つけることができます。

更に、全ての構成パラメタはメイングループおよびサブグループの中でソートされます。メイングループは、構成パラメタが属するアプリケーションを意味します。例えば中央のOTRSフレームワーク用「フレームワーク」、チケット・システムの「チケット」、FAQシステム用の「FAQ」などです。アプリケーションがグループ・リストボックスから選択され、「グループ選択」ボタンが押される場合、サブグループにアクセスすることができます。

すべての構成パラメタはチェックボックスからオン・オフできます。パラメタをオフにすれば、システムはこのパラメタを無視するかまたはデフォルトを使用します。リセット用リンクを使用して、変更した構成パラメタをシステム・デフォルトへ戻し切り替えることは可能です。アップデート・ボタンですべてのシステム構成パラメタへの変更を送信します。

例えば
　迅速に新規インストールをセットアップ等、システムの構成に加えた変更をすべて保存したい場合には「エクスポート設定」ボタンを使用できます。その際、.pmファイルを作成します。自身の設定をリストアするためには、「インポート設定」ボタンを押すだけです。そして作成した.pmファイルを選択します。

注記

セキュリティ上の理由で、データベース接続用の構成パラメタはSysConfigセクションの中で変更することができません。Kernel/Config.pmの中でそれらを手動でセットしなければならない。

3. システムのバックアップ

本章はバックアップとOTRSデータのリストアについて記述します。

3.1. バックアップ

バックアップする2つのタイプのデータがあります:ひとつはアプリケーションファイル(例えば　/opt/otrs の中のファイル)、もうひとつはデータベースに格納されたデータです。

バックアップを単純化するために、スクリプト　scripts/backup.pl　はOTRSのインストレーションで含まれています。それは全てのインポートデータをバックアップするように走らせることができます。(下記のスクリプトを参照)。

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

スクリプト：*OTRS*バックアップメカニズムに関するヘルプ

バックアップを作成するために以下のスクリプトで指定したコマンドを実行します：

```
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#
```

スクリプト:バックアップの作成

全てのデータはディレクトリ /backup/2010-09-07_14-28/　に格納されます。（下図参照）それに加えてデータはa .tar.gz fileに保存されます。

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

スクリプト:バックアップ・ファイルのチェック。

3.2. リストア

バックアップをリストアするために、保存されたアプリケーション・データはインストレーション・ディレクトリ（例えば、/opt/otrs　）に書き戻されなければいけません。さらに、データベースをリストアしなければなりません。

リストア・プロセスを簡素化するスクリプト　scripts/restore.pl　(下記のスクリプトを参照)は、OTRSの全ての出荷時に同梱され、MySQLとPostgreSQLをサポートします。

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

スクリプト:リストア・メカニズムに関するヘルプ

例えば保存されるデータ（例えば、ディレクトリー/backup/2010-09-07_14-28/の中にある）は、以下のスクリプトで指定されたコマンドでリストアすることができます。（OTRSは/opt/otrsにインストールされていると仮定します。）

```
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#
```

スクリプト: *OTRSデータのリストア*

4. Eメール設定

4.1. Eメールの送信/受信

4.1.1. Eメールの送信

4.1.1.1. Sendmail(デフォルト)経由。

OTRSは、Sendmail、Postfix、QmailあるいはEximによってEメールを送信することができます。デフォルト構成はsendmailを使用し使用開始時に即稼働します。

グラフィカルな構成フロント・エンドによってsendmailの設定を構成することができます。(Framework::Core::Sendmail)

4.1.1.2. SMTPサーバまたは smarthost経由。

OTRSは、SMTP(Simple Mail Transfer Protocol / RFC 821)か安全なSMTPによってEメールを送信できます。非UNIXプラットフォーム(例えばウィンドウズ)上でこれを使用したいと思うでしょう。

SMTPサーバー・セッティングは、SysConfig(Framework::Core::Sendmail)によって構成できます。オプションのSMTPSを利用可能として見つけることができないならば、必要なパール・モジュールがないということです。その場合は、"OTRSに必要なパール・モジュールのインストレーション"を参照してください。

4.1.2. Eメールの受信

4.1.2.1. OTRS GUI OTRSで構成されたEメールアカウント

OTRSは、POP3、POP3S、IMAPおよびIMAPSメールアカウントからEメールを受け取ることができます。

管理者ページのPostMasterメールアカウント・リンク経由でメールアカウントを構成して下さい。

新規メールアカウントが作成されることになっている場合(下記の図を参照)そのメールサーバ名、ログイン名およびパスワードが指定されなければいけません。また、POP3, POP3S, IMAP あるいは IMAPS等のようなメールサーバタイプを選択する必要があります。サーバー・タイプのオプションとして利用可能なものが見あたらなければ、システムに必要なPerlモジュールがないことがその理由です。その場合、指示に関しては "OTRSに必要なパール・モジュールのインスタレーション"を参照してください。

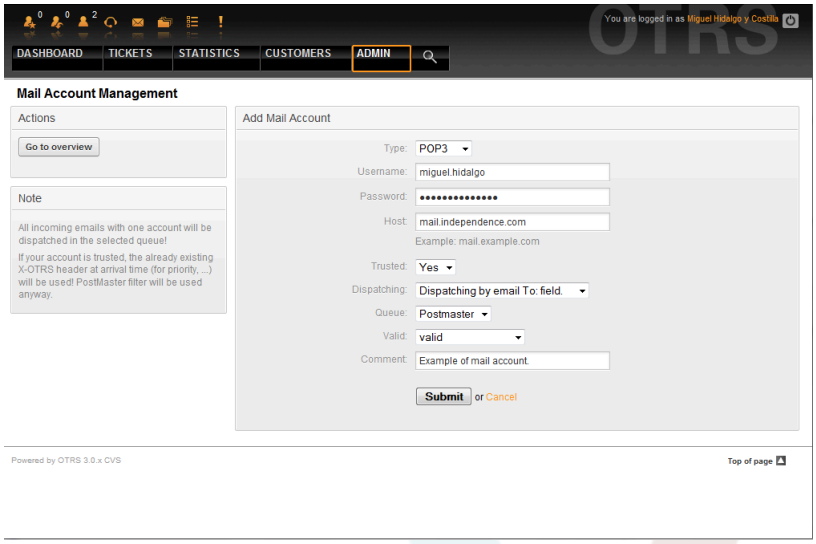


図:メールアカウントの追加。

信頼済オプションの値にYesを選べば、受信メッセージに付けられたどんなX-OTRSヘッダーも評価され実行されます。X-OTRSヘッダーがチケットシステムの中のいくつかのアクションを実行することができるので既知の送信者だけに信頼済オプションをYesにセットするべきです。X-OTRSヘッダーはOTRSではフィルタ・モジュールによって使用されます。X-OTRSヘッダーは、このテーブルの中でより詳細に説明されます。作成されたどんなpostmasterフィルタ規則も、信頼されたオプションのセッティングに関係なく実行されます。

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.

メールアカウント用データはすべてOTRSデータベースの中で保存されます。`otrs.PostMasterMailbox.pl` スクリプト(OTRSインストレーションの`bin`ディレクトリーにあります)は、データベースの中で設定を使用し、メールを取って来ます。全てのメール設定が適切に動いているかどうかをチェックするために`./bin/otrs.PostMasterMailbox.pl`を手動で実行することができます。

正常なインストレーションにおいては、`postmaster_mailbox` cronジョブによってメールが10分ごとに取って来られるでしょう。cronジョブの修正に関するさらに詳しい情報に関しては、"「OTRSのためにcronジョブをセット・アップする」" セクションを参照してください。

注記

メールを取って来る場合、OTRSはPOPかIMAPサーバーからメールを削除します。さらにサーバ上でコピーを保存するオプションはありません。サーバのコピーを保持したければ、メールサーバで転送規則を作成するとよいでしょう。詳細のためにメールサーバ・ドキュメンテーションを調べてください。

4.1.2.2. コマンドライン・プログラムおよびprocmail経由 (otrs.PostMaster.pl)

OTRSにEメールを入れるためにメールアカウントを使用することができなければ、コマンドライン・プログラム `bin/otrs.PostMaster.pl`はその問題を解決するひとつの方法かもしれませ

ん。それはSTDIN経由でメールを取り、OTRSにそれらを直接送ります。それは、MDA(メール配信エージェント(例えばprocmail))がこのプログラムを実行すれば、EメールがOTRSシステムにおいて利用可能だろうということを意味します。

MDAなしで bin/otrs.PostMaster.plをテストするには、次のスクリプトのコマンドを実行してください。

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

スクリプト:MDAのないポストマスターのテスト

EメールがQueueViewの中で表示される場合、あなたの設定は正しく動作しています。

ProcmailはLinux環境中でとても普及しているEメール・フィルタです。ほとんどのシステムにそれがインストールされます。そうでなければ、*procmail*ホームページを見てください。

OTRS用にprocmailを構成するためには(sendmail、postfix、eximあるいはqmailといったprocmail構成のMTAに基づいた)、~otrs/.procmailrc.distファイルを使用して、.procmailrcにそれをコピーして、スクリプトのラインを下に加えてください。

```
SYS_HOME=$HOME
PATH=/bin:/usr/bin:/usr/local/bin
# --
# 全てのEメールをthe PostMaster processへPipe します。
# --
:0 :
| $SYS_HOME/bin/otrs.PostMaster.pl
```

スクリプト:OTRSのための*procmail*の構成。

localのOTRSユーザへ送られたEメールはすべて、bin/otrs.PostMaster.plに送られ、QueueViewで表示されるでしょう。

4.1.2.3. otrs.PostMaster.plのための**POP3, IMAP**および**fetchmail**によるEメールの取得

メールサーバから、POP3またはIMAPのメールボックス経由で、OTRSマシン/ローカルのOTRSアカウント、およびprocmailにEメールを受け取るために、fetchmailを使用します。

注記

OTRSマシン上で作動してるSMTP構成が必要です。

OTRSのホームディレクトリーの中で.fetchmailrc.distを使用し、必要に応じて.fetchmailrc.Modfiy/changeにそれをコピーすることができます。(下記の例7-1を参照)

例**4.2 .fetchmailrc**

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

.fetchmailrcを710("chmod 710.fetchmailrc")にセットすることを忘れないでください!

上記の例7-1からの.fetchmailrcで、コマンド**fetchmail -a**が実行されれば、Eメールはすべてローカルの**OTRS**アカウントへ転送されるでしょう。メールを規則的に取って来たい場合は、このコマンドでcronjobをセット・アップしてください。

4.1.2.4. OTRS/PostMasterモジュール(より多く複雑な送信用の)によるフィルター/送信

bin/otrs.PostMaster.plあるいはbin/otrs.PostMasterMailbox.plの方法を使用すれば、PostMasterフィルタ・モジュールを備えたX-OTRSヘッダー・エントリを挿入するか修正することができます。チケットシステムは、X-OTRSヘッダーによって、受信に対するいくつかのアクションを実行するか、特定のキューへそれらをソートするか、優先度を変更するか、あるいは例えば顧客IDを変更することができます。X-OTRSヘッダーに関してより多くの情報、OTRS管理者ページからの「メールアカウントの追加」に関するセクションにおいて利用可能です。

いくつかのデフォルト・フィルタ・モジュールがあります:

> **注記**
>
> ジョブ名 (e.g. $Self->{'PostMaster::PreFilterModule'}->{'JobName'}) はユニークである必要があります。

ernel::System::PostMaster::Filter::Match は、いくつかのEメールヘッダ(e.g. From, To, Subject, ...)をマッチさせるためのデフォルトモジュールです。 フィルタ規則がマッチすれば、それは新規Eメールヘッダ(例： X-OTRS-Ignore: yes or X-OTRSQueue:spam)にセットできます。例 7-2のジョブは Kernel/Config.pm に挿入可能です。

例4.3 フィルタ・モジュールのためのサンプルジョブ

```
# Job Name: 1-Match
# (noreply@からの全てのスパムEメールをブロックまたは無視)
$Self->{'PostMaster::PreFilterModule'}->{'1-Match'} = {
    Module => 'Kernel::System::PostMaster::Filter::Match',
    Match => {
        From => 'noreply@',
    },
    Set => {
        'X-OTRS-Ignore' => 'yes',
    },
};
# Job Name: 2-Match
# (件名: **ORDER**かつsales@example.comからのEメールをソート)
# 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は外部コマンドにEメールを送る、デフォルト・モジュールです。出力はSTDOUTに与えられます、また、結果がTrueの場合、新規Eメール・ヘッダー(例えば、X-OTRS-Ignore: yes or X-OTRS-Queue: spam) をセットします。例7-3は Kernel/Config.pmで使用することができます。

例4.4 フィルタ・モジュールのためのサンプル・ジョブ **Kernel::System::PostMaster::Filter::CMD**

```
# Job Name: 5-SpamAssassin
# (SpamAssassin サンプルセットアップ, スパムメールを無視します)
$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は、Eメールの件名、本文、または正規表現を使用する両方に、外部識別子を解析する可能性を加えるデフォルト・モジュールです。その後、それは定義された動的フィールドにこの値を格納します。Eメールが中へ入る時、OTRSは最初に外部識別子を検索し、1つ見つけると、あらかじめ定義された動的フィールドについてOTRSにクエリします。それは既存のチケットを見つければ、このチケットを更新するでしょう。そうでなければ、それは別々のフィールドに外部参照番号で新規チケットを作成するでしょう。

OTRS SysConfigは、異なる外部チケット番号を設定するために４つの異なる設定を既に供給していますより多くのセッティングが必要な場合、それらを手動で加える必要がある必要があります。次の例はSysConfig設定を拡張するためにKernel/Config.pmの中で使用することができます。

例4.5 フィルター・モジュールのためのサンプル・ジョブ Kernel::System::PostMaster::Filter::ExternalTicketNumberRecognition

```
    # Job Name: ExternalNumber
# 外部チケット番号の認識では、<送信者>@externalticket.comのアドレス
# からの入信Eメールの中の件名と本文インシデント<番号>をチェックします。
# <番号>が見つかればそれは動的フィールド「ExternalNumber(外部番号)」
# に保存されます。（管理者パネルで設定が必要）
    $Self->{'PostMaster::PreFilterModule'}->{'000-ExternalTicketNumberRecognition5'} =  {
        'FromAddressRegExp' => '\\s*@externalticket.com',
        'NumberRegExp'      => 'Incident-(\\d.*)',
        'SearchInSubject'   => '1',
        'SearchInBody'      => '1',
        'TicketStateTypes'  => 'new;open'
        'DynamicFieldName'  => 'ExternalNumber',
        'Module'            =>
 'Kernel::System::PostMaster::Filter::ExternalTicketNumberRecognition',
        'Name'              => 'Test External Ticket Number',
        'SenderType'        => 'system',
    };
```

構成オプション

- FromAddressRegExp

  これはオプションの設定です。この「From:」アドレスと一致しているメールだけがこのフィルターで考慮されます。この設定を調整して、送信メール用外部システム利用を、送信者アドレスにすることができます。このアドレスが異なる場合、このオプションを空にすることができます。この場合、OTRSは送信者アドレスをチェックしません。

- NumberRegExp

  これは必須設定項目です。この設定は
  　件名またはチケットの本文からチケット番号を抽出するためにOTRSが使う正規表現を含みます。デフォルトの正規表現は
  　以下のような例の発生と一致するでしょう。例えば「Incident-12354」そしてその部分を動的フィールド中に括弧で挟む部分を置きます（ここでは「12354」）

- SearchInSubject

  これが「1」にセットされる場合、Eメール件名はチケット番号を求めて検索されます。

- SearchInBody

  これが「1」にセットされる場合、Eメール本文はチケット番号を求めて検索されます。

- TicketStateTypes（チケット・状態・タイプ）

これはオプションの設定です。もし与えられれば、それは与えられた状態タイプの開いた外部チケットのみOTRSを検索するでしょう。状態タイプはセミコロンで分離されます。

- DynamicField

  これは必須の設定で、外部番号を保存するために使われている動的フィールドを定義します。(フィールド名はシステムに存在しなければいけないしかつ有効でなければなりません)

- SenderType

  これは、OTRSで記事作成に使用される送信者タイプを定義します。

もちろん、自分のPostMasterフィルタ・モジュールを開発することも可能です。

4.2. PGP鍵を備えた安全なEメール

OTRSは、PGP鍵を備えた送信メッセージに署名するか暗号化する能力を持っています。更に、暗号化された受信メッセージは解読することができます。暗号化と解読はGPLツールGnuPGで実施されます。OTRSのためのセットアップGnuPGに、次のステップを行なわなければなりません。

1. オペレーティング・システムのパッケージ・マネージャーによるGnuPGのインストール
2. OTRSで使うGnuPGを構成してください。GnuPGと秘密鍵のための必要なディレクトリを作成しなければなりません。シェルからユーザ「otrs」としてスクリプト中で下に示されるコマンドを実行しなければなりません。

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

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

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

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

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

Passphrase: secret
Repeat passphrase: secret

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

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

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

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

linux:~$
```

スクリプト: *GnuPG*の構成

スクリプトの中で下に示されるように、デフォルト構成は、ほとんどの要求されるパラメタのために適用することができます。キーに指定された適切なパスワードで、key所有者に対する値だけを正確に入力しなければなりません。

3. PGP　の設定画面では、PGPはOTRSで有効になっており(一つ目のオプション)、また、PGPプログラムへのパスが設定され、有効になっている必要があります

   次の構成設定(PGP::Options)はさらに変わることを要求するかもしれません。このコンフィグ・セッティングによって、「otrs」ユーザによってgpgのすべての実行に使用されるパラメタは指定することができます。特に「otrs」ユーザのGnuPGのための構成ファイルのディレクトリは重要です。例/opt/otrs/.gnupgの中で、使用されます。このディレクトリは、PGP鍵構成以前に作成されました。

   次の構成オプション(PGP::Key::Password)によって、キーIDのペアおよび自身の秘密鍵のパスワードを指定することは可能です、なぜなら、外部からのコミュニケーションパートナーチケットシステムに彼らのメッセージを暗号化と秘密鍵で書き込み、OTRSはそれらのメッセージをIDとパスワードをここで指定して解読するからです。

   どのようにしてあなた自身の秘密鍵のIDを得ることができますか？あなた自身の秘密鍵のIDはすでにキー生成時に表示されています。（上記ステップ1をご覧ください）さらに、次のスクリプト中で指定されたコマンドがユーザ「otrs」として実行される場合、IDを得ることが可能です:

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

linux:~$
```

スクリプト:自分の秘密鍵の*ID*を得ること。

秘密鍵のIDは、「サブ」で始まるラインで見つけることができます。それは長さ8文字で16進法の文字列で上の例では「52B97069」です。チケットシステムでこのキー用に指定しなければならないパスワードは、キー生成中に与えられたのと同じです。

このデータが挿入された後、「アップデート」ボタンはセッティングを保存するために使用することができます。これでOTRSは暗号化されたメッセージを受け取り解読する準備ができています。

4. このデータが挿入された後、「アップデート」ボタンはセッティングを保存するために使用することができます。これでOTRSは暗号化されたメッセージを受け取り解読する準備ができています。

   最初の方法は、顧客管理インターフェースで顧客の公開鍵を指定することです。

   第2の方法は、PGP鍵設定(管理者ページから到達可能)によってキーを指定することです。この画面の右側のセクションにおいては、すでにインポート済の顧客の公開鍵が全て表示されます。PGP鍵がOTRSのために活性化され構成されたら、自分の公開鍵もそこにリストされるべきです。PGP鍵設定画面の左のエリアでキーを探せます。さらに、新規公開鍵はファイルからシステムへアップロードすることができます。

   OTRSへインポートされる必要のある公開鍵を備えたファイルはGnuPGP互換性をもつキー・ファイルでなければなりません。ほとんどの場合、ファイルに格納されたキーは「ASCII armored key」です。OTRSはこのフォーマットを取り扱えます。

4.3. S/MIMEを備えた安全なEメール

一見、S/MIMEを備えた暗号化はPGP鍵を備えた暗号化より少し複雑に見えます。最初に、OTRSシステムのために認証局(CA)を設立しなければなりません。後のステップは、PGP鍵で必要とされるものに非常に似ています:OTRSの構成、自身の証明書のインストール、必要な他の証明書のインポート等。

S/MIME構成は、大部分はOTRSウェブインターフェースの外部で行われ、「otrs」ユーザによってシェルの中で実行されるべきです。Linux下のMIME構成SSL(OpenSSL)に基づきます。したがって、あなたのシステムにOpenSSLパッケージがインストールされるかどうかを第一にチェックしてください。OpenSSLパッケージは、CA.pl(証明書生成の最も重要なステップはそれで行なうことができます)と呼ばれるスクリプトを含んでいます。手続きを単純化するために、ファイルシステムでCA.plスクリプトが保存され、シェルのパス変数に一時的に位置を入力する場所を見つけます。(下記のスクリプトを参照)

```
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>
```

スクリプト:S/*MIME*の構成

上記のスクリプトは、新規一時ディレクトリ~/tmpが作成されたことを表示しています。証明書はその中で生成されることになっています。

証明書を作成するためには、コマンドライン中から次のオペレーションを行なってください:(我々は、OTRS管理者がテスト用のSSL証明書および学習する目的を作成しなければならないと考えます。暗号化用の保証されたSSL証明書を既に持った場合それを使用して、これらのステップをスキップしてください。)

1. SSLのために自分の認証局を開設してください。自分のSSL証明書の要請を証明するためにそれを必要とする(下記のスクリプトを参照)。

```
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>
```

スクリプト:　*SSL*のために認証局を設立

2. 証明書リクエストを生成してください(下記のスクリプトを参照)

```
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>
```

スクリプト: 証明書リクエストの作成

3. 証明書リクエストは署名され、それによってあなた自身のCAに証明されます。あるいは別の外部の保証されたCAによって署名されることによりさらに信頼性を高められます。（下記スクリプト参照）

```
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>
```

スクリプト:証明書リクエストの署名

4. あなた自身の証明書を生成して下さい。そしてそれに伴い、署名された証明書を使う、すべてのデータを生成して下さい。（下記スクリプト参照）

```
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>
```

スクリプト:新規証明書の生成

今、これらのオペレーションが実行され、S/MIMEセットアップはOTRSの中で完成したはずです。

セットアップのこの部分はリンク"SMIME"を選んで、管理者ページで実行されます。OTRSで一般的なS/MIMEサポートがまだ可能にされてない場合、マスク(mask)は管理者にこれを指摘し、それを可能にするために適切なリンクを提供します。

SysConfig group で"Crypt::SMIME" S/MIME に関する全般の設定が行えます

ここで、S/MIME支援を活性化し、OpenSSLコマンド用のパスおよび証明書用のディレクトリーを定義することができます。上に作成されたキー・ファイルは、ここで示されたディレクトリーに格納されなければいけません。そうでなければ、OpenSSLは使用できません。

次のステップはOTRS管理ページ上のS/MIME構成で行われます。ここで、OTRSシステムの秘密鍵および他のコミュニケーション・パートナーの公開鍵(複数)をインポートすることができます。このセクションの最初に作成し、OTRSに追加した公開鍵を入力してください。

言うまでもなく、コミュニケーション・パートナーの全てのパブリックS/MIMEキーは顧客管理ツールを使用してインポートすることもできます。

5. 外部バックエンドの使用

5.1. 顧客情報

OTRSは、ユーザー名、メール・アドレス、電話番号などのような多くの顧客情報属性で動作しています。これらの属性は担当者および顧客フロント・エンドの両方に表示され、顧客の認証に使用されます。

OTRSの内に使用されるか表示された顧客情報は高度にカスタマイズ可能です。しかし次の情報は、顧客認証に常に必要とされます:

- User login（ユーザーログイン）
- Email address（Eメールアドレス）
- Customer ID（顧客ID）

担当者インターフェースの顧客情報を表示したければ、次のSysConfigパラメタを使用して下さい。

```
# Ticket::Frontend::CustomerInfo*
```

```
    # （作成（電話とEメール）、ズーム、キュービュー時に顧客情報を表示）
    $Self->{'Ticket::Frontend::CustomerInfoCompose'} = 1;
    $Self->{'Ticket::Frontend::CustomerInfoZoom'} = 1;
```

スクリプト：*SysConfig*構成パラメタ

5.2. 顧客ユーザ・バックエンド

2つのタイプの顧客バックエンド、DBおよびLDAPを使用することができます。あなたが別の顧客バックエンド(例えばSAP)を既に持っていれば、それを使用するモジュールを書くことはもちろん可能です。

5.2.1. データベース(デフォルト)

例11-1は、DB顧客バックエンドの構成を示します。それは、OTRSデータベースに保存された顧客情報を使用します。

例4.6 DB顧客バックエンドの構成

```
# CustomerUser (customer database backend and settings)
$Self->{CustomerUser} = {
    Name => 'Database Datasource',
    Module => 'Kernel::System::CustomerUser::DB',
    Params => {
       # 外部データベースを使用したければ必要な設定を追加します。
#            DSN => 'DBI:odbc:yourdsn',
#            Type => 'mssql', # only for ODBC connections
#            DSN => 'DBI:mysql:database=customerdb;host=customerdbhost',
#            User => '',
#            Password => '',
            Table => 'customer_user',
            # フロント・エンドがunicodeで顧客データベースサーバのcharsetが
            # iso-8859-1である場合は、これらのオプションを使用して下さい。
#           SourceCharset => 'iso-8859-1',
#           DestCharset => 'utf-8',

# CaseSensitive（大文字と小文字を区別せず評価するかどうかの指定値）は、
# 大文字小文字の評価を鈍感に動かすためにSQLステートメントが
# LOWER()関数呼び出しを必要とするかどうかをコントロールします。
# これを1にセットすると、大きなデータベース上でパフォーマンスを
# 劇的に改善するでしょう。
            CaseSensitive => 0,
        },
# 顧客固有識別ID
CustomerKey => 'login',

# 顧客 #
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,
#    # 顧客パネルに自身のチケットを非表示とします, CompanyTicketsn#
 CustomerUserExcludePrimaryCustomerID => 0,
#    # 自動ログインを生成します
#    AutoLoginCreation => 0,
#    AutoLoginCreationPrefix => 'auto',
#    # admin は顧客の選択（選好）を変更できます。
#    AdminSetPreferences => 1,
#    # キャッシュの生存時間（秒）- どのデータベース・クエリーのキャッシュでも
#    CacheTTL => 0,
#    # 読み取り専用のソース
```

```
#    ReadOnly => 1,
    Map => [
        # ノート：ログイン、Eメールと顧客IDが必要！
        # 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 ],
    ],
    # デフォルトの選択
    Selections => {
        UserTitle => {
            'Mr.' => 'Mr.',
            'Mrs.' => 'Mrs.',
        },
    },
};
```

顧客情報をカスタマイズしたい場合は、カラム・ヘッダーを変更するか、あるいは新規ものをOTRSデータベース中のcustomer_userテーブルに加えてください。例として、下記のスクリプトは、room番号用の新規フィールドを加える方法を教えます。

```
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:~#
```

スクリプト: *room*フィールドの*customer_user*テーブルへの追加。

次のスクリプトで示されるように、さて、新規カラムをKernel/Config.pm中のMAP配列に追加してください。

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

```
    [ 'UserRoom',      'Room',      'room',       0, 1, 'var', '', 0 ],
```

スクリプト:roomフィールドの*Kernel/Config.pm*ファイルへの追加。

担当者インターフェースの顧客リンクによってこの顧客情報をすべて編集することも可能です。

5.2.1.1. 多数のID(会社チケット)を持った顧客

顧客に1つを超える顧客IDを割り当てることは可能です。顧客が他の顧客のチケットへアクセスしなければならないならばこれは役立ちます。(例えば、スーパーバイザはアシスタントのチケットを見たい。)顧客が他の顧客のチケットへアクセスできるならば、OTRSの会社チケット機能が使われます。会社チケットは、顧客パネル中の「会社チケット」リンクによってアクセスすることができます。

会社チケットを使用するために、顧客にとってアクセス可能であるべきIDを備えた新規カラムをOTRSデータベース中のcustomer_userテーブルに加えなければなりません(下記のスクリプトを参照)。

```
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:~#
```

スクリプト: *customer_ids*フィールドの*customer_user*テーブルへの追加。

スクリプトの中で下に示されるように、新規カラムをKernel/Config.pm中のMAP配列に加えなければなりません。

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

スクリプト: *UserCustomerIDs*フィールドの *Kernel/Config.pm*ファイル への追加。

ここで、多数の顧客IDのための新規カラムは、顧客管理のためのセクションの中で担当者・インターフェース経由で編集することができます。

顧客が他の顧客のチケットにアクセスすることを確実にするために、多数の顧客IDのための新規フィールドの中へのこれらの他のユーザのIDを追加します。セミコロンによって各IDを区切って下さい(下記の例11-2を参照)。

例4.7 DBバックエンドを備えた会社チケットの使用

顧客A、BおよびCはあなたのシステムに存在します。また、Aは顧客パネルからBとCのチケットにアクセスしたい。BとCは他のユーザのチケットにアクセスするべきではありません。

このセットアップを実現するためには、上に記述されるようなKernel/Config.pm中のcustomer_userテーブルおよびMappingを変更してください。次に、担当者インターフェースあ

るいは管理者ページの顧客リンクから、顧客Aのための設定をロードしてください。設定が表示される場合、CustomerIDsのためのフィールドへ値「B;C;」を追加して下さい。

5.2.2. LDAP

顧客情報を備えたLDAPディレクトリーを持っているならば、例11-3において示されるように、OTRSを備えた顧客バックエンドとしてそれを使用することができます。

例4.8 LDAP顧客バックエンドの構成

```
# CustomerUser
# (顧客LDAPバックエンドとその設定)
$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',
        # 次の記述は有効です。しかし、単に、匿名ユーザが
        # LDAPツリーから読み込み許可を持っていない場合のみに必要です。
        UserDN => '',
        UserPw => '',
        # 常にあるフィルタを各LDAPクエリに追加したい場合このオプションを使います。
        # このオプション、例えば AlwaysFilter => '(mail=*)' or AlwaysFilter =>
 '(objectclass=user)'
        AlwaysFilter => '',
            # フロントエンドとLDAPの両方がunicodeなら、これを使います:
            SourceCharset => 'utf-8',
            DestCharset   => 'utf-8',
            # フロントエンドがunicodeでLDAPサーバが
            # iso-8859-1ならば、このオプションを使います。
            # SourceCharset => 'iso-8859-1',
            # DestCharset => 'utf-8',
            # Net::LDAP new params (必要なら詳細はperldoc Net::LDAPを参照)
            Params => {
                port => 389,
                timeout => 120,
                async => 0,
                version => 3,
            },
    },
    # 顧客識別ID
    CustomerKey => 'uid',
    # 顧客 #
    CustomerID => 'mail',
    CustomerUserListFields => ['cn', 'mail'],
    CustomerUserSearchFields => ['uid', 'cn', 'mail'],
    CustomerUserSearchPrefix => '',
    CustomerUserSearchSuffix => '*',
    CustomerUserSearchListLimit => 250,
    CustomerUserPostMasterSearchFields => ['mail'],
    CustomerUserNameFields => ['givenname', 'sn'],
    # 顧客パネルに自身のチケットを非表示, CompanyTickets
    CustomerUserExcludePrimaryCustomerID => 0,
    # 有効なユーザー用にLDAPフィルタを追加 (上級者向け設定)
#    CustomerUserValidFilter => '(!(description=locked))',
    # administrator は顧客選択 ( 選好 ) を変更できません。
    AdminSetPreferences => 0,
#    # キャッシュ生存時間 ( 秒 ) - cache any database queries
#    CacheTTL => 0,
    Map => [
        # ノート : ログイン、Eメールと顧客IDは必須!
        # 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 ],
    ],
};
```

追加の顧客属性（マネージャーの名前、携帯電話番号あるいは部署のような）が、あなたのLDAPディレクトリーに格納されていて、この情報をOTRSに表示したいならば、次のスクリプトの中で示されるように、これらの属性へのエントリーを備えたKernel/Config.pm 中のMAP配列をただ拡張してください。

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

スクリプト:新規フィールドの*Kernel/Config.pm*ファイルへの追加

5.2.2.1. 多数のID(会社チケット)を持った顧客

LDAPバックエンドを使用する場合、顧客にひとつ以上の顧客IDを割り当てることが可能です。会社チケットを使用するために、新規フィールドを顧客によってアクセス可能なIDを含んでいるLDAPディレクトリーに加えなければなりません。

LDAPディレクトリーの新規フィールドが作成されている場合、スクリプトの中で下に示されるように、新規エントリーをKernel/Config.pm中のMAP配列に加えなければなりません。

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

スクリプト:　新規フィールドの*Kernel/Config.pm*ファイルへの*Maping*

複数の顧客IDのためのフィールドをLDAPディレクトリーで直接編集しなければなりません。OTRSはLDAPから単に読み込むだけで書き込むことはできません。

顧客によって他の顧客のチケットにアクセスを保証するためには、そのチケットがあなたのLDAPディレクトリーの新規フィールドにアクセスされるべき顧客の顧客IDを加えてください。各IDはセミコロンで区切って下さい(下記の例11-4を参照)。

例4.9 LDAPバックエンドを備えた会社チケットの使用

顧客A、BおよびCはあなたのシステムに存在します。また、Aは顧客パネルからBとCのチケットにアクセスしたい。BとCは他のユーザのチケットにアクセスするべきではありません。

このセットアップを実現するためには、上に記述されるようなKernel/Config.pm中のLDAPディレクトリーおよびMappingを変更してください。その後、CustomerIDsのためのフィールドへ値「B; C;」をあなたのLDAPディレクトリーの顧客Aのために追加して下さい。

5.2.3. OTRSでひとつ以上の顧客バックエンドを使用する

あなたがOTRS(例えばLDAPおよびデータベース・バックエンド)と共に使用されるひとつ以上の顧客情報ソースを利用したければ、CustomerUser構成パラメタを「数」で(例えば「CustomerUser1」、「CustomerUser2」)拡張するべきです。(下記の例11-5を参照)。

例4.10 OTRSでひとつ以上の顧客バックエンドを使用する

以下は、OTRSでLDAPとデータベース顧客バックエンドの両方を使用する構成例です。

```
# 1. Customer user backend: DB
# (顧客データベース・バックグラウンドと設定)
$Self->{CustomerUser1} = {
    Name => 'Customer Database',
    Module => 'Kernel::System::CustomerUser::DB',
    Params => {
        # 外部データベースを使用したければ
        # 必要な設定を追加して下さい。
#        DSN => 'DBI:odbc:yourdsn',
#        Type => 'mssql', # only for ODBC connections
#        DSN => 'DBI:mysql:database=customerdb;host=customerdbhost',
#        User => '',
#        Password => '',
        Table => 'customer_user',
    },
    # customer unique id
    CustomerKey => 'login',
    # customer #
    CustomerID => 'customer_id',
    CustomerValid => 'valid_id',
    CustomerUserListFields => ['first_name', 'last_name', 'email'],
    CustomerUserSearchFields => ['login', 'last_name', 'customer_id'],
    CustomerUserSearchPrefix => '',
    CustomerUserSearchSuffix => '*',
    CustomerUserSearchListLimit => 250,
    CustomerUserPostMasterSearchFields => ['email'],
    CustomerUserNameFields => ['title','first_name','last_name'],
    CustomerUserEmailUniqCheck => 1,
#    # show not own tickets in customer panel, CompanyTickets
#    CustomerUserExcludePrimaryCustomerID => 0,
#    # generate auto logins
#    AutoLoginCreation => 0,
#    AutoLoginCreationPrefix => 'auto',
#    # admin can change customer preferences
#    AdminSetPreferences => 1,
#    # cache time to live in sec. - cache any database queries
#    CacheTTL => 0,
#    # just a read only source
#    ReadOnly => 1,
    Map => [

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

```
    },
};

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

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

10までの異なる顧客バックエンドを統合することは可能です。全ての顧客情報を見るか編集する(書き込みアクセス権限が可能と仮定して)ためにOTRSの中で顧客管理インターフェースを使用してください。

5.2.4. 動的フィールドにCustomerUserデータを格納すること

顧客データを直接チケットの動的フィールドに格納することも時には役立つかもしれません。(例えばこのデータで特別な統計を作成するような目的で)

チケットが作成される場合、あるいはチケットの顧客が変更される場合、動的フィールド値がセットされます。動的フィールドの値は顧客データから得られます。これはすべてのバックエンドのために働きますが、特にLDAPバックエンドに役立ちます。

OTRSのこのオプション機能を活性化するために、以下の二つの設定を活性化して下さい。「Ticket::EventModulePost###930-DynamicFieldFromCustomerUser」および「DynamicFieldFromCustomerUser::Mapping」。後の設定では、CustomerUserフィールド入力がチケットのどの動的フィールドに格納しなければならないかの構成を含んでいます。フィールドはシステムの中にあるに違いありませんし、AgentTicketFreeTextのために可能になるに違いありません。そのゆえ、それらは手動でセットすることができます。それらはAgentTicketPhone、AgentTicketEmailおよびAgentTicketCustomerのために可能になってはなりません。もしそれらが可能ならば、それらは自動設定値に対して先行するでしょう。

5.3. 担当者と顧客を認証するバックエンド

OTRSは、異なるバックエンドに対する、担当者および顧客を認証するオプションを提示します。

5.3.1. 担当者のための認証バックエンド

5.3.1.1. DB(デフォルト)

デフォルトで使用される担当者を認証するバックエンドは、OTRSデータベースです。担当者は、管理者ページの担当者管理インターフェイス経由で 追加・編集されます(下のサンプル11-6参照)。

例**4.11 DB**バックエンドに対する認証エージェント

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

5.3.1.2. LDAP

LDAPディレクトリが担当者データをすべて保存する場合、OTRSの中のユーザを認証するためにLDAPモジュールを使用することができます(下記の例11-7を参照)。このモジュールはLDAPツリーに対して単に読み込みアクセスを有しています。それは、担当者管理インターフェース経由で利用者データを編集することができないことを意味します。

例**4.12 LDAP**バックエンドに対する認証エージェント

```
# これはLDAP認証バックエンド用サンプル構成です。
# (Net::LDAPがインストール済であることを確認下さい。)
$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';

# ユーザがposixGroupで認証を許可されていることを確認します。
# (例えばユーザはOTRSを利用するグループ xyz に属する必要があります。)
$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';

# 次は有効です。しかし  匿名ユーザがLDAPツリーから
# 読み込み許可を持たない場合のみ必要です。
$Self->{'AuthModule::LDAP::SearchUserDN'} = '';
$Self->{'AuthModule::LDAP::SearchUserPw'} = '';

# 常にあるフィルタを各LDAPクエリに追加したい場合このオプションを使います。
# たとえば AlwaysFilter => '(mail=*)' or AlwaysFilter => '(objectclass=user)'
$Self->{'AuthModule::LDAP::AlwaysFilter'} = '';

# ログイン名のに接頭辞を追加したければ、このオプションを使います。
# たとえばユーザがuserを使いたいが、LDAPにuser@domainが存在する場合。
#    $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,
};
```

スクリプトの中で下に示される構成パラメータは、ローカルのOTRSデータベースの中へのLDAPディレクトリーからの利用者データを同期させるために使用することができます。これは、LDAPサーバーにリクエストの数を減らし、OTRSでの認証の速度を向上させます。担当者が最初に認証する場合、データ同期は終っています。ローカルのOTRSデータベースへデータを同期することができますが、LDAPディレクトリーは認証のための最後の実体（インスタンス）です、したがって、LDAPツリー内の活性化されていないユーザは認証することができません。アカウント・データがOTRSデータベースに既に保存されている場合さえ。LDAPディレクトリーの担当者データは、OTRSのウェブインターフェース経由で編集することができません。したがって、LDAPツリーにデータを直接管理しなければなりません。

```
# AuthModule用にAuthSyncBackend (AuthSyncModule)定義します。
# キーが存在しそれが空ならば、Syncにならないでしょう。
# サンプル値: AuthSyncBackend, AuthSyncBackend2
$Self->{'AuthModule::UseSyncBackend'} = 'AuthSyncBackend';

# LDAPに対する担当者データ sync
$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
# (最初の担当者のログインの初期の作成後に、rw許可を持つsync followingグループ)
$Self->{'AuthSyncModule::LDAP::UserSyncInitialGroups'} = [
    'users',
];
```

スクリプト：*OTRS*データベースの中への*LDAP*ディレクトリーからの利用者データを同期させること。

5.3.1.3. 担当者のための**HTTPBasicAuth**

すべての担当者のための「シングル・サインオン」解決を実行したければ、OTRSのためにHTTP基本認証(すべてのシステム用の)およびHTTPBasicAuthモジュールを使用することができます(下記の例11-8を参照)。

例**4.13 HTTPBasic**を使用して、担当者を認証します。

```
# これはapache ($ENV{REMOTE_USER})認証バックエンド用のサンプル構成です。
# apache http-basic-authのシングル・サインオンを持ちたければそれを使います。
$Self->{'AuthModule'} = 'Kernel::System::Auth::HTTPBasicAuth';

# Note:
#
# このモジュールを使うなら、代替手段として次の構成設定を使うべきです。
# (もしユーザがapache ($ENV{REMOTE_USER})として認証されていないならば)
$Self->{LoginURL} = 'http://host.example.com/not-authorised-for-otrs.html';
$Self->{LogoutURL} = 'http://host.example.com/thanks-for-using-otrs.html';
```

5.3.1.4. Radius

例11-9において示される構成パラメータは、Radiusサーバーに対する担当者を認証するために使用することができます。

例**4.14 Radius**バックエンドに対する担当者を認証します。

```
# これは、radiusサーバに対するauth. agentsへのサンプル構成です。
$Self->{'AuthModule'} = 'Kernel::System::Auth::Radius';
$Self->{'AuthModule::Radius::Host'} = 'radiushost';
$Self->{'AuthModule::Radius::Password'} = 'radiussecret';
```

5.3.2. 顧客のための認証バックエンド

5.3.2.1. データベース(デフォルト)

OTRSの中の顧客のための既定のユーザー認証バックエンドはOTRSデータベースです。このバックエンドで、顧客情報はすべて、OTRSのウェブインターフェース経由で編集することができます(下記の例11-10を参照)。

例**4.15 DB**バックエンドに対する顧客ユーザ認証

```
# これはthe otrs dbに対する認証モジュールです。
$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

すべての顧客データを備えたLDAPディレクトリーを持っていれば、OTRSへの顧客を認証するためにLDAPモジュールを使用することができます(下記の例11-11を参照)。このモジュールがLDAPバックエンドに対して単に読み取り専用アクセスをしているので、OTRSウェブインターフェース経由で顧客情報を編集することはできません。

例**4.16 LDAP**バックエンドに対する顧客ユーザ認証

```
# これはLDAP認証バックエンド用サンプル構成です。
# Net::LDAPがインストールされていることを確認下さい。)
$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';

# ユーザがposixGroupで認証することが可能かどうか確認します。
# ( 例えばユーザはOTRSを使うにはグループ xyzに属する必要があります。 )
$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';

# 次は有効です。しかし　匿名ユーザがLDAPツリーから
# 読み込み許可を持たない場合のみ必要です。
$Self->{'Customer::AuthModule::LDAP::SearchUserDN'} = '';
$Self->{'Customer::AuthModule::LDAP::SearchUserPw'} = '';

# 常にあるフィルタを各LDAPクエリに追加したい場合このオプションを使います
# このオプションは例えば
# AlwaysFilter => '(mail=*)' or AlwaysFilter =>
'(objectclass=user)'です。
$Self->{'Customer::AuthModule::LDAP::AlwaysFilter'} = '';

# ログイン名のに接頭辞を追加したければ、このオプションを使います。
# たとえばユーザがuserを使いたいが、LDAPにuser@domainが存在する場合。
#$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. 顧客のための**HTTPBasicAuth**

全顧客ユーザに「シングル・サイン・オン」ソリューションを実行したければ、HTTPBasic認証（あなたの全てのシステムに対して）を使い、そしてOTRSのHTTPBasicAuthモジュールを使う（OTRSでログインが必要なくなります）。下記の例11-12を参照してください。

例**4.17 HTTPBasic**を備えた顧客ユーザ認証

```
# apache ($ENV{REMOTE_USER})認証バックエンド用のサンプル構成です。
# apache http-basic-authのシングル・サインオンを
# 持ちたければそれを使います。
$Self->{'Customer::AuthModule'} = 'Kernel::System::CustomerAuth::HTTPBasicAuth';

# ノート:
# このモジュールを使うなら、代替手段として次の構成設定を使うべきです。
# もしユーザがapache ($ENV{REMOTE_USER})として認証されていないならば
$Self->{CustomerPanelLoginURL} = 'http://host.example.com/not-authorised-for-otrs.html';
$Self->{CustomerPanelLogoutURL} = 'http://host.example.com/thanks-for-using-otrs.html';
```

5.3.2.4. Radius

例11-13において示される設定はRadiusサーバーに対するあなたの顧客を認証するために使用することができます。

例**4.18 Radias**のバックエンドに対する顧客ユーザ認証

```
# これはRaduisサーバへの認証顧客へのサンプル構成です。
$Self->{'Customer::AuthModule'} = 'Kernel::System::Auth::Radius';
$Self->{'Customer::AuthModule::Radius::Host'} = 'radiushost';
$Self->{'Customer::AuthModule::Radius::Password'} = 'radiussecret';
```

5.4. 顧客自己登録のカスタマイズ

新規顧客(customer.plパネルによってアクセス可能)のための自己登録をカスタマイズすることは可能です。新規の任意または必須フィールド（room番号、アドレスあるいは状態のような）を追加することができます。

次の例は、顧客のroom番号を格納するためには、この場合どのように顧客データベースの中であなたが必須フィールドを指定することができるか示します。

5.4.1. ウェブインターフェースのカスタマイズ

customer.plウェブインターフェースのroom番号用の新規フィールドを表示するために、このインターフェースのレイアウトの責任者である.dtlファイルを修正しなければなりません。80行目のまわりの新規フィールドを加えて、Kernel/Output/HTML/Standard/CustomerLogin.dtlファイルを編集してください。(下記のスクリプトを参照)。

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

スクリプト:ウェブインターフェースでの新規フィールドの表示

5.4.2. 顧客マッピング

次のステップで、顧客マッピングはroom番号への新規エントリーで拡張しなければなりません。変更がアップデート後に失われないことを確実にするためには、Kernel/Config.pm に Kernel/Config/Defaults.pmからの「CustomerUser」設定を入れてください。スクリプト中で下に示されるように、MAP配列を変更して、新規room番号フィールドを追加してください。

```
# CustomerUser
# （顧客データベース・バックエンドと設定）
$Self->{CustomerUser} = {
    Name => 'Database Backend',
    Module => 'Kernel::System::CustomerUser::DB',
    Params => {
        # 外部データベースを使いたければ、
        # 必須設定を追加する必要があります。
#        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,
#    # 顧客パネルで自身のチケットではなく顧客チケットを表示します。
```

```
#    CustomerUserExcludePrimaryCustomerID => 0,
#    # 自動ログインを生成します。
#    AutoLoginCreation => 0,
#    AutoLoginCreationPrefix => 'auto',
#    # 管理者は顧客の選択 ( 選好 ) を変更できます。
#    AdminSetPreferences => 1,
#    # キャッシュが生存する秒数 - キャッシュ・データベース必須
#    CacheTTL => 0,
#    # 読み込み専用のソース
#    ReadOnly => 1,
    Map => [

        # ノート : ログイン、Eメール、顧客IDは必要です！
        # 変数、フロントエンド、保存場所、表示(1=常時,2=軽度)、必須、
        # 保存タイプ、httpリンク、読み込み専用、httpリンク・ターゲット。
        [ '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.',
        },
    },
};
```

スクリプト: マッピング配列の変更

5.4.3. OTRS DBでcustomer_userテーブルのカスタマイズ

最後のステップは、新しいroom番号カラムをOTRSデータベース中customer_userテーブルに追加することです(下記のスクリプトを参照)。このカラムでは、room番号へのエントリーは格納されます。

```
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:~#
```

スクリプト:新しいカラムを*customer_user*テーブルに追加

room用の新規フィールドに入力されたら、顧客情報パネルに表示され、顧客・ユーザ管理画面にも表示されます。さらに、新しいアカウントを登録する場合、新しい顧客はroom番号を挿入しなければなりません。マイクロソフトのIISの上でOTRSを使用する場合、`Config.pm`の中で行なわれた変更を活性化するためにウェブサーバを再起動するべきです。

6. チケット設定

6.1. チケットの状態

6.1.1. 事前定義された状態

OTRSでは、事前定義されたチケット状態とタイプを変更または新規追加することができます。状態名と状態タイプという2つの属性が状態にとって重要です。

OTRSのデフォルト状態は以下の通りです。「完了（成功）」('closed successful')、「完了（不成功）」('closed unsuccessful')、「結合済」('merged')、「新規」('new')、「対応中」('open')、「保留（自動完了+）」('pending auto close+')、「保留（自動完了-」('pending auto close-')、「保留（期限付）」('pending reminder')、「削除」('removed'.)。

6.1.1.1. 「新規」（New）

チケットは、受信Eメールから作成された時、通常この状態です。

6.1.1.2. 対応中（open）

これはキューと担当者に割り当てられたチケットのためのデフォルト状態です。

6.1.1.3. 保留（期限付）

保留時間が終了すると、チケット所有者はチケットに関するリマインダーEメールを受信します。チケットがロックされていなければ、リマインダーはキュー中の全担当者へ送られます。リマインダーチケットは単に業務時間に送信され、チケット状態が担当者によって変更されるまで、24時間ごとに繰り返し送信されます。このステータスでのチケットによって費やされた時間は、さらにエスカレーション時間として追加計上されます。

6.1.1.4. 保留（自動完了-）

保留時間が終了したらこの状態のチケットは「完了（不成功）」('closed unsuccessful')にセットされます。このステータスでのチケットによって費やされた時間は、さらにエスカレーション時間として追加計上されます。

6.1.1.5. 保留（自動完了+）

保留時間が終了したらこの状態のチケットは「完了（成功）」('closed successful')にセットされます。この状態のチケットに費やされた時間は、さらにエスカレーション時間として追加計上されます。

6.1.1.6. 結合済

これは他のチケットと結合されたチケットの状態です。

6.1.1.7. 完了（成功）

これはうまく解決されたチケットの最終状態です。構成に依存しますが、一旦完了したチケットを再対応可能にも不可能にも構成可能です。

6.1.1.8. 完了 (不成功)

これは解決できなかったチケット用の最終状態です。構成に依存しますが、一旦完了したチケットを再対応可能にも不可能にも構成可能です。

6.1.2. 状態のカスタマイズ

すべての状態は名(状態名)およびタイプ(状態タイプ)を持っています。管理ページの状態リンクをクリックします。そして新規状態を作成する"状態追加"ボタンを押します。新しい状態の名前は自由に選択することができます。ウェブインターフェイス経由では状態タイプを変更することはできません。 新規タイプを追加あるいは既存名を変更したい場合、データベース直接変更されなければいけません。概してデフォルトの状態タイプは変更すべきではありません。予測不能の結果を生むことがあるためです。例えば、エスカレーション計算とアンロック機能は特別な状態タイプに基いています。

既存の状態名は変更することができます。あるいは、新規状態はこの画面から追加されます。「新規」状態が、ウェブインターフェース経由で変更されたなら、この変更も、構成ファイル Kernel/Config.pmまたはSysConfigインターフェイス経由で構成変更されねばなりません。以下のスクリプトで指定された設定は、「新規」に変更された状態で、OTRSが確実に稼働するように変更されなければなりません。

```
    [...]
    # PostmasterDefaultState
    # (新規チケットのデフォルト状態)  [default: new]
    $Self->{PostmasterDefaultState} = 'new';

    # CustomerDefaultState
    # (新規顧客チケットのデフォルト状態)
    $Self->{CustomerDefaultState} = 'new';
    [...]
```

スクリプト:Kernel/*Config.pm*の設定修正

万一新規状態タイプが加えられるならば、OTRSデータベースにあるthe ticket_state_typeテーブルは、下記のスクリプトで表示されるように、データベース・クライアント・プログラムで変更される必要があります。

```
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:~#
```

スクリプト:OTRSデータベースの修正

あなたが今作成した新規状態タイプを使用することは可能である。状態がこの新規状態タイプとリンクされた後、OTRS構成も、さらに新規状態が使用可能であることを確実にするために変更されなければならない。SysConfigにて次のオプションを修正してください:

Ticket -> Frontend::Agent::Ticket::ViewPhoneNew > AgentTicketPhone###StateDefault - to define the default next state for new phone tickets.

Ticket -> Frontend::Agent::Ticket::ViewPhoneNew > AgentTicketPhone###StateType - to define the available next states for new phone tickets.

Ticket -> Frontend::Agent::Ticket::ViewEmailNew > AgentTicketEmail###StateDefault - to define the default next state for new email tickets.

Ticket -> Frontend::Agent::Ticket::ViewEmailNew > AgentTicketEmail###StateType - to define the available next states for new email tickets.

Ticket -> Frontend::Agent::Ticket::ViewPhoneOutbound > AgentTicketPhoneOutbound###State - to define the default next state for new phone articles.

Ticket -> Frontend::Agent::Ticket::ViewPhoneOutbound > AgentTicketPhoneOutbound###StateType - to define the available next states for new phone articles.

Ticket -> Frontend::Agent::Ticket::ViewMove > AgentTicketMove###State - to define the default next state for moving a ticket.

Ticket -> Frontend::Agent::Ticket::ViewMove > AgentTicketMove###StateType - to define the available next states for moving a ticket.

Ticket -> Frontend::Agent::Ticket::ViewBounce > StateDefault - to define the default next state after bouncing a ticket.

Ticket -> Frontend::Agent::Ticket::ViewBounce > StateType - to define the available next states in the bounce screen.

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

Ticket -> Frontend::Agent::Ticket::ViewBulk > StateType - to define the available next states in the bulk action screen.

Ticket -> Frontend::Agent::Ticket::ViewClose > StateDefault - to define the default next state after closing a ticket.

Ticket -> Frontend::Agent::Ticket::ViewClose > StateType - to define the available next states in the close screen.

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

Ticket -> Frontend::Agent::Ticket::ViewCompose > StateType - to define the available next states in the Compose (reply) screen.

Ticket -> Frontend::Agent::Ticket::ViewForward > StateDefault - to define the default next state after forwarding a ticket.

Ticket -> Frontend::Agent::Ticket::ViewForward > StateType - to define the available next states in the Forward screen.

Ticket -> Frontend::Agent::Ticket::ViewForward > StateDefault - to define the default next state of a ticket in the free text screen.

Ticket -> Frontend::Agent::Ticket::ViewForward > StateType - to define the available next states in the free text screen.

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

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

Ticket -> Core::PostMaster > PostmasterFollowUpStateClosed - to define the state of tickets after a follow-up has been received on an already closed ticket.

Ticket -> Core::Ticket > ViewableStateType - to define the state types that are displayed at various places in the system, for example in the Queueview.

Ticket -> Core::Ticket > UnlockStateType - to define the state types for unlocked tickets.

Ticket -> Core::Ticket > PendingReminderStateType - to define the state type for reminder tickets.

Ticket -> Core::Ticket > PendingAutoStateType - to define the state type for Pending Auto tickets.

Ticket -> Core::Ticket > StateAfterPending - to define the state a ticket is set to after the Pending Auto timer of the configured state has expired.

6.2. チケット・プロパティ

OTRSには、 5 つのデフォルトの優先度レベルが付属しています。それは、管理者ページにある「プロパティ」"Priorities"
経由で変更可能です。カスタマイズされた優先度リストを作成する場合、ユーザ・インターフェース中の優先度選択箱の中でそれらがアルファベット順にソートされると心に留めておいてください。さらに、OTRSはQueueViewの中の内部データベースIDによってチケットを順序づけします。

注記

他のOTRS実体でそうであるように、優先度は、削除できないかもしれませんし、有効オプションを無効まはた無効-暫定にする設定によって、単に非活性化されるかもしれません。

重要

新規優先度が加えられた場合、あるいは既存のものが変更された場合、SysConfigの中のいくつかの値を修正したいと思うかもしれません:

- Ticket:Core::Postmaster::PostmasterDefaultPriority - defines the default priority for all incoming emails.
- Ticket:Frontend::Agent:Ticket::ViewPhoneNew:Priority - defines the default priority in the New Phone Ticket screen for agents.
- Ticket:Frontend::Agent:Ticket::ViewEmailNew:Priority - defines the default priority in the New Email Ticket screen for agents.
- Ticket:Frontend::Customer:Ticket::ViewNew:PriorityDefault - defines the default priority in the New Ticket screen in the Customer frontend.

6.3. チケット責任者とチケットの監視

OTRS 2.1からチケットの所有者に加えて、 1 チケットに対して 1 名の責任者を割り当てることが可能です。さらに、チケットと関係のある全ての活動はチケットの所有者以外の誰かに見られるようにできます。これらの2つの機能性は、チケット責任者 (TicketResponsible) とチケット監視者

（TicketWatcher）としてインプリメントされ、タスクの割り当ておよび階層的チーム構造内に働くことを手助けします。

6.3.1. チケット責任者

チケット責任者の機能は、チケット所有者以外に担当者によってチケットの完全な処理を手助けします。したがってチケットをロックした担当者はそれをもう一人の担当者に渡すことができます。渡された担当者はチケットの所有者ではなく、顧客依頼に対する第二の返答者となります。依頼が取り扱われた後、最初の担当者は第2の担当者からチケット責任を譲り受けできます。

With the configuration parameter Ticket::Responsible, the ticket responsibility feature can be activated. This will cause 3 new links to appear in the ticket activities menu of a zoomed ticket in the agent interface.

チケット責任者は、チケット内容を呼び出して割り当てることができます。そしてチケット活動メニュー「責任者」リンク（担当者インターフェイスにあるズームチケットにある）をクリックします。（下図参照）

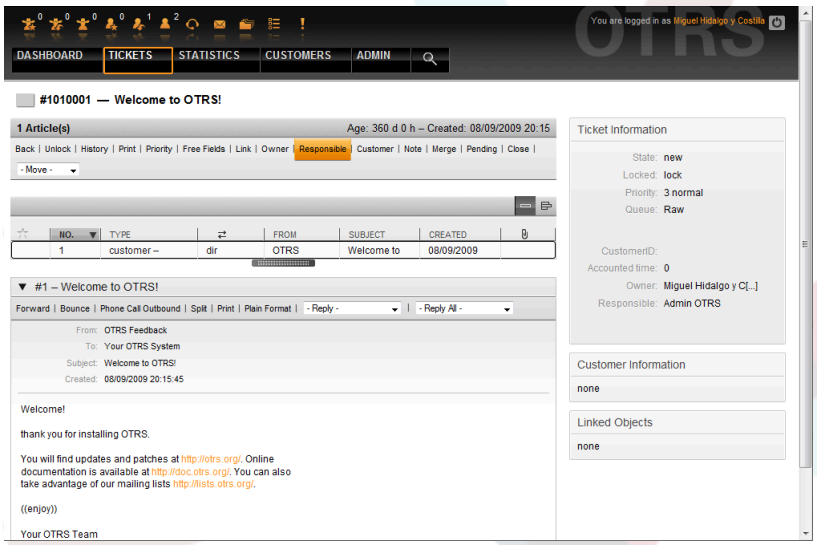
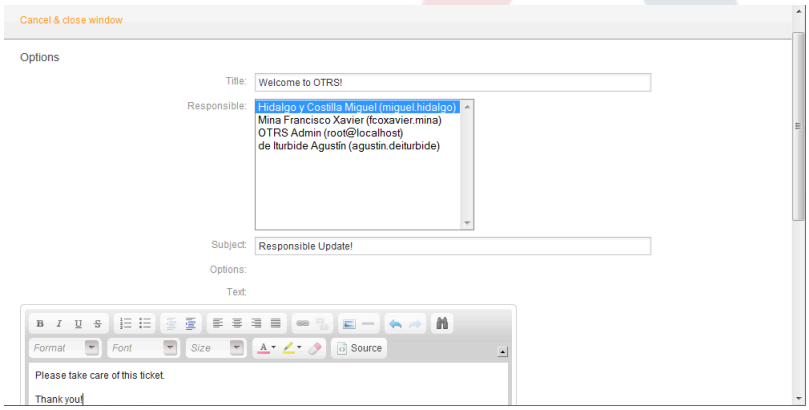


図:ズーム・ビューでのチケット責任者の変更

「責任者」をクリックしたのちに、チケットの責任者変更ダイアログのポップアップが開きます。（下図参照）このダイアログは新規責任者の担当者へメッセージを送信するために使用することもできます。



図：チケット責任者を変更するポップアップ・ダイアログ

チケット責任者機能が活性化されるとすぐに、担当者が責任を負う全チケットのリストは、OTRS担当者インターフェイスの責任者ビューを通してアクセスできます。

6.3.2. チケットの監視

OTRS2.1から　「チケット監視」機能を使えば
　スーパバイザのような担当者を選ぶことにより、システム内の特定のチケットを処理することなく監視することができます。

The TicketWatcher feature can be activated with the configuration parameter Ticket::Watcher which adds new links to your actions toolbar. Using Ticket::WatcherGroup, one or more user groups with permission to watch tickets can also be defined.

チケットを見るためには、ズーム・ビューに行って、チケット活動メニュー中の「購読」リンクをクリックしてください(下記の図を参照)。

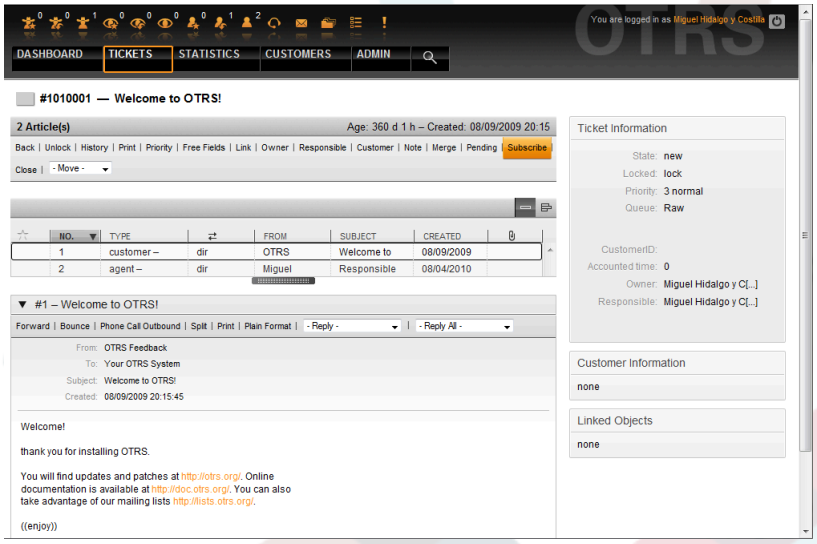


図:ズーム・ビューでチケットを監視するために購読する

もう特定のチケットを見たくない場合は、ズーム・ビューへ行き、チケット活動メニュー中の「購読しない」リンクをクリックします(下記の図を参照)。

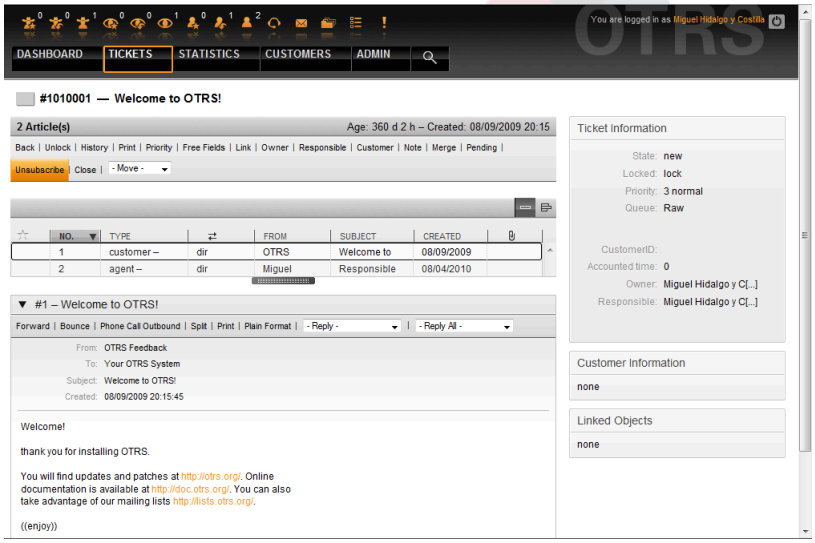


図:ズーム・ビューでチケットの監視の登録解除

監視中のすべてのチケットのリストは、チケット監視者機能が活性化されるとすぐに、OTRS担当者インターフェイス（下図参照）の監視ビュー経由でアクセス可能になりました

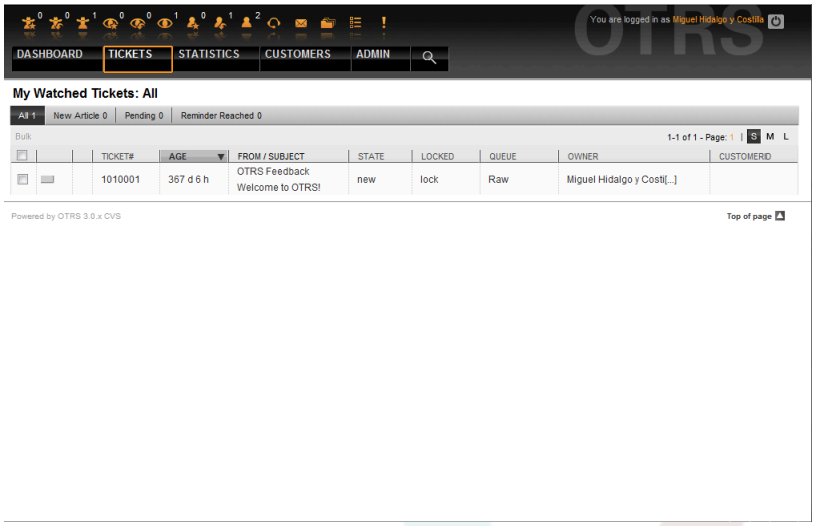


図:監視チケットビュー

7. 時間関連機能

7.1. 業務時間、祝祭日とタイムゾーンの設定

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. 業務時間

Set up the working hours for your system in SysConfig Framework > Core::Time::TimeWorkingHours, or for your specific calendar in the calendar's configuration. OTRS can handle a granularity of one hour. Checking the marks in the boxes 8, 9, 10 ... 17 corresponds with business hours of 8 AM - 6 PM.

業務時間内にのみエスカレーション・チケットを作成することができます。エスカレーションの通知や送信された保留チケットロックされていないチケットも同様です。

7.1.2. 固定の期日、祝祭日

Holidays that are on a fixed date every year, such as New Year's Day or the Fourth of July, can be specified in TimeVacationDays, or in the corresponding section for the calendars 1-9.

チケットはエスカレーションしませんしまた、TimeVacationDaysに定義された日付にアンロックされません。

注記

OTRS出荷のデフォルトでドイツの祝祭日をインストールしています。

7.1.3. TimeVacationDaysOneTime

Holidays such as Easter that do not have a yearly fixed date but instead vary each year, can be specified in TimeVacationDaysOneTime.

チケットはエスカレーションしませんしまた、TimeVacationDaysに定義された日付にアンロックされません。

> **注記**
>
> OTRSは、一度だけの休日をつけて出荷しません。イースターまたは感謝祭のような休日はOTRSを構成する際にご自身でシステムに追加して下さい。

7.2. 自動アンロック

ロックされたチケットは、システムによって自動的に対応中にすることができます。この機能はこんな場合に有用でしょう。例えば、担当者が処理される必要のあるチケットをロックしているが、彼が何らかの理由でそれらを処理できなければ、「彼は今緊急の用事のためオフィスの外にいます」ということができます。自動アンロック機能は、ロックされたチケットがないことが忘れられないことを確実にするために与えられた時間の後でチケットをアンロックします。

チケットが対応中にされる前の時間の量は、すべてのキューのためのキュー設定の中で指定することができます。モジュール `bin/otrs.UnlockTickets.pl`（cronジョブとして周期的に実行される)は、チケットの自動アンロックします。アンロック中のチケットに関する通知は以下の担当者のみに送信されます。

「マイ・キュー」にあるアンロックされたチケット・セットを持つキューを持っている担当者。個人的選択（選好）のアンロック・チケットに関する通知を活性化した担当者。

次の条件がすべて満たされれば、チケットはアンロックされます:

- チケットがあるキューのために定義されたアンロック・タイムアウトがあります。
- チケットはロック済にセットされています。
- チケットの状態は対応中です。

アンロック・タイマーは担当者がチケットに新しい外部記事を追加すればリセットされます。それは次のタイプのうちのいずれかでしょう。:　外部*E*メール、電話、ファックス、*sms*あるいは外部ノート。

さらに、チケット中の最後の記事が担当者によって作成され、顧客が別のものを加えれば、ウェブあるいはEメール応答のいずれかによって、アンロック・タイマーはリセットされるでしょう。

アンロック・タイマーをリセットする最後のイベントは、チケットがもう一人の担当者にいつ割り当てられるかです。

8. PDF出力のカスタマイズ

このセクションは、OTRSの中のPDF出力に対する構成可能なオプションを扱います。

If you use the Print action from anywhere within the OTRS interface, it will generate a formatted PDF file. You can deactivate this by modifying the configuration parameter PDF to create HTML output instead.

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.

Perl　CPANモジュールPDF::API2およびCompress::Zlibはpdfファイルの生成のためにインストールされなければいけません。多くのディストリビューションでは、それらはパッケージとして利用可能で、それぞれのパッケージ・マネージャーを使用して、容易にインストールすることができます。これが可能ではなかった場合、CPANでそれらをインストールしなければならない。Perlモジュールのインストールに関するさらに詳しい情報に関しては、"Perlモジュールの設置"セクションを参照してください。

9. 統計モジュール (Stats module)

OTRS統計モジュールは、運用上の統計を追跡するための機能を保持し、OTRS使用量に関連したカスタムレポートを生成します。OTRSシステムは、様々な指標を示すレポートに言及するために用語「stat」を属に関して使用します。

OTRS統計モジュールの適切な構成は多くの必要条件および考察に関連しています。これらは以下を含んでいます。様々な評価すべきOTRSモジュール、ユーザ許可設定、計算指標とそれらの複雑さのレベル、統計モジュールの構成の簡易化・軽減、計算の高速化・効率化、豊富な出力セットのサポート。

統計要素(つまり特定の必要条件用の統計モジュールの機能性を補足するファイル)は、複雑な統計の計算のために統合することができます。

9.1. 担当者によるモジュールの取り扱い

担当者として署名すると、図中のように、ナビゲーション・バーは、様々なサブメニュー・オプションと共に、リンク「統計」を表示します。

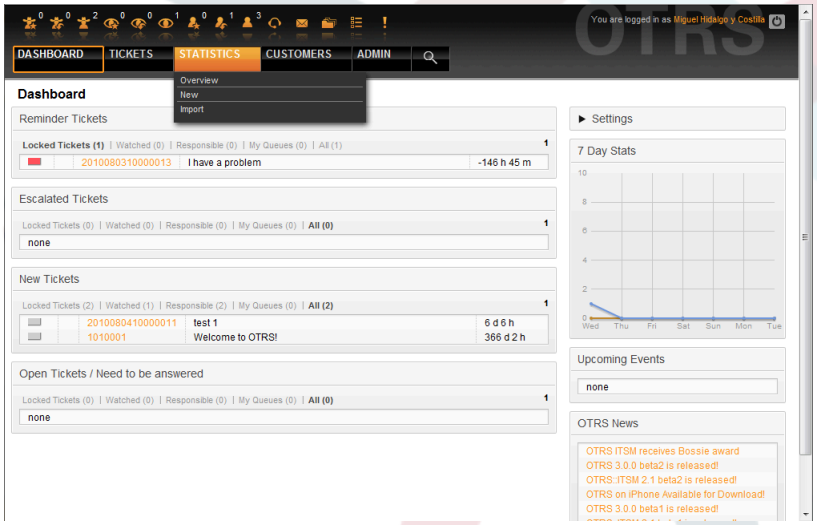


図:統計メニュー・オプション

統計メニューの中で提供される異なるオプションは次のとおりである:

- 概要。異なる事前構成されたレポートのリストを示します。
- 新規。rw権限を必要とします。
- インポート。rw権限を必要とします。

9.1.1. 概要

ナビゲーション・バー中の「統計」リンクと、サブメニュー・リンクの「概要」を選択することで、概要画面を呼び出します。概要画面は、全ての事前定義された担当者が利用可能なレポートをのリストを表します(下記の図を参照)。

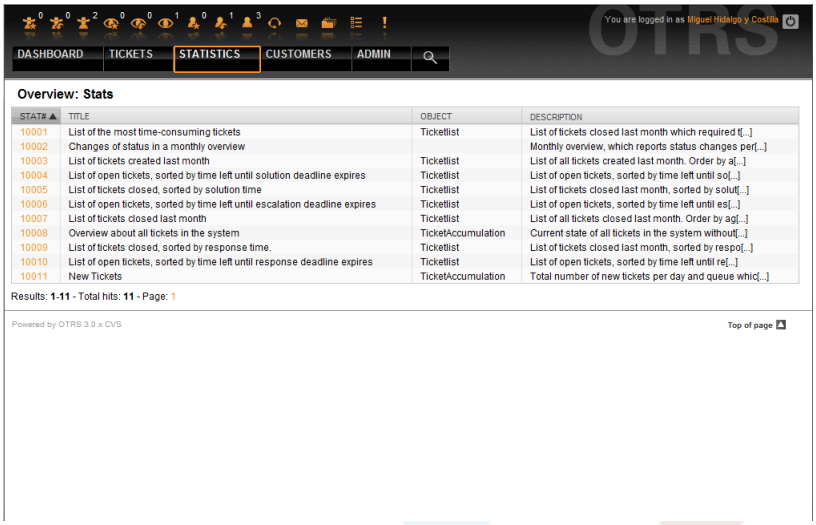


図:標準レポートの概要

次の情報が概要にリストされたそれぞれの標準レポートのために提供される:

- *Stat#*。ユニークなレポート番号。
- *Title*。レポートのタイトル。
- オブジェクト。オブジェクトは統計生成するために使用します。静的統計の場合には、動的オブジェクトがその生成に使用されないとともに、オブジェクトは表示されません。
- 摘要：レポートの簡潔な記述。

統計モジュールがインストールされている場合、それはシステムへインポートされた少数のサンプルのレポートでプレインストールされたようになる。これらは概要ページのリストとして示されます。概要リストが単一のページを越えるものまで及ぶ場合、担当者は異なるページをブラウズすることができます。レポートのリストは、リスト中の希望のカラム・ヘッダーのクリックすることで、好みに応じてソートすることができます。特別のレポートを生成するためには、概要リスト中でレポートに関連したstat番号をクリックしてください。これによりレポート用の「ビュー」インターフェイスが表示されます。

9.1.2. レポートの生成と閲覧

ビュー・ユーザ・インターフェースはstatの構成設定を提供します(下記の図を参照)。

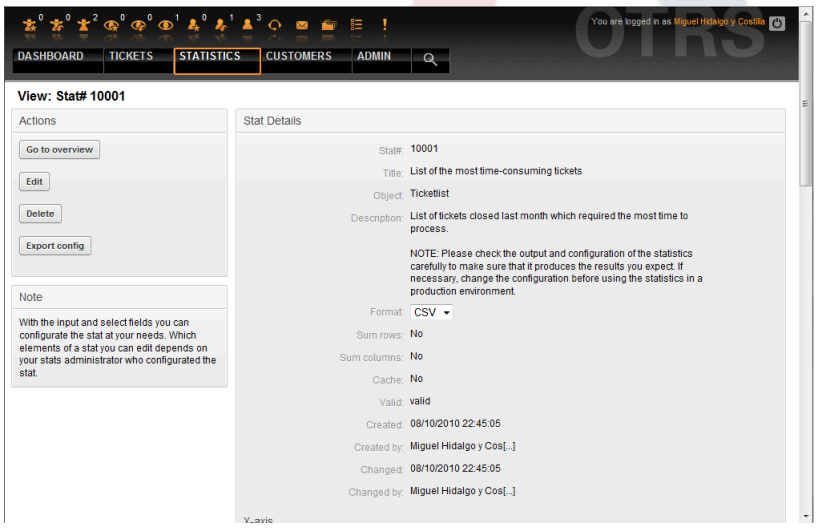


図:特定のレポートのビュー

特定のレポート用構成設定は
ビュー画面の中のオプションの範囲の中で設定できます。レポート製作者あるいは適切な許可を持つ他の誰でも、設定を作ることができます。

下記参照:

- 可能なアクション:
  - 概要に行ってください。レポートの概要リストにリンクしてください。
  - 編集。現在のレポート構造(必要なrw権)を編集してください。)。
  - 削除。現在のレポート(要求されるrw権)を削除してください。
  - 構成のエクスポート。ファイル・ダウンロード(必要なrw権)によって、レポート構成をエクスポートしてください。

    使用法:エクスポート・インポート機能で、テストシステム上でのレポートのテストと作成が便利になります。そしてそれに続く生産システムへの統合を容易にします。
- レポート詳細:
  - *Stat#*。レポートの数。
  - *Title*。レポートのタイトル。
  - オブジェクト。レポートの生成のために使用されたオブジェクト。
  - 摘要。レポートの目的を記載するための摘要。
  - フォーマット。構成に依存しますが、次いずれかの出力フォーマットによるレポート出力フォーマット:
    - CSV。
    - 印刷。
    - グラフ・線。
    - グラフ・棒。
    - グラフ・横棒。
    - グラフ・点。
    - グラフ・点ー線。
    - グラフ・エリア。
    - グラフ・パイ。
  - グラフ化。グラフィック/図表用ピクセルでのサイズ。レポート構成が図表を許可する場合のみ、このオプションが与えられます。一般に使用可能なグラフィックのサイズはすべて
    SysConfigの中のOTRS管理者によって構成されます。レポートを構成する一方で、担当者はあらかじめ適切なフォーマットをすべて選択することができます。
  - 行の合計。レポートが、列(複数のセルが各行の合計を表している)によって修正されるかどうかを示します。
  - 列の合計。レポートが、行(複数のセルが各列の合計を表している)によって修正されるかどうかを示します。

- キャッシュ。生成されたレポートがファイルシステムにキャッシュされるかどうかを示します。
- 有効。どんな理由でもレポートを一時的に実行してはならない場合これは「無効」にセットすることができます。そうすると、パネルの右側底部にある「スタート」ボタンは、もはや表示されません。そのレポートはもはや生成することができません。
- 作成済。レポートの生成時間。
- 作成済。レポートを作成した担当者の名前。
- 変更済。レポートが最後に修正された時刻による。
- 変更済。レポートを最後に変更した担当者の名前による。
- *X*軸。この機能を使用すると、担当者はxとyの軸(OTRS管理者によって活性化された時だけ)を切り替えることができます。
- レポート自体に関する情報は一般情報に続きます。2つの異なるレポート(あるいはstat)ビューがあります:
  - 静的stat・ビュー。静的レポート生成者は、統計モジュールへ統合することができます(下記の図を参照)。

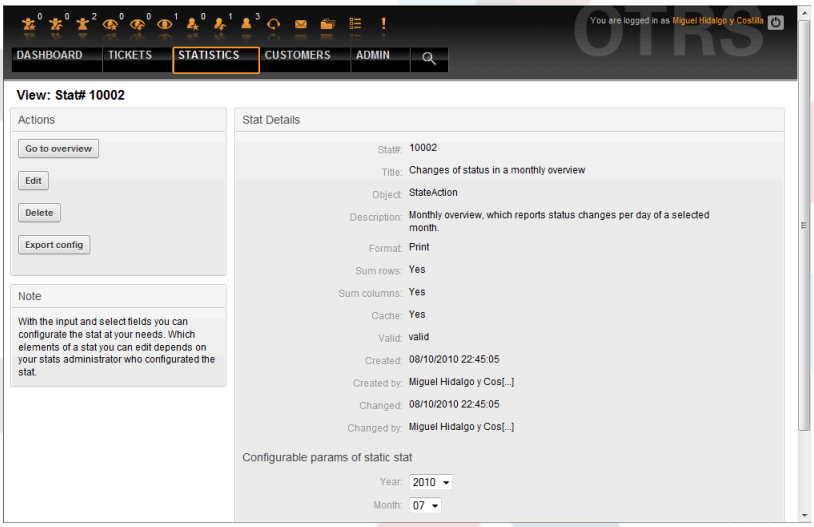


図:静的レポートのビュー

  - 動的統計*(stat)*ビュー(上記の図を参照)。それらは2つの異なる方法で表示することができる:
    - 不変の設定。レポートの作成者は、このフィールドを変更する許可を持っていません。
    - 可変の設定。そのようなレポートの構成設定は担当者によって変更することができます。

「スタート」ボタン(画面の底の)を押すことは、レポートを生成する最後のステップです。これら二つのボタンが表示されない理由の可能性は二つあります。:

1. そのレポートは無効にセットされ、それゆえ非活性化されます。
2. そのレポートはクリーンに構成されなかったため、まだ実行可能ではありません。この場合、必要な情報はOTRS通知セクション(ナビゲーション・バーより下の)で見つけることができます。

ビューページのセッティングが正しくない場合、「スタート」ボタンが押された後、このページは再び示されます。また、どの入力が正しくなかったかに関する情報は通知セクションの中で提供されます。

9.1.3. 編集/新規

書き込み権を持った担当者は、統計モジュールの編集ユーザ・インターフェースを呼び出すことにより、既存のレポート構成を編集することができます。一方、彼らは新規レポート作成できます。関連する画面に、次の方法で到達することができます:

1. 編集:統計ビュー中の「編集」ボタン経由
2. 新規:ナビゲーション・バーからの統計メニュー中の「新規」リンク、あるいは概要ページからの「追加」ボタン経由。

統計は4ステップのウイザードで編集されます:

1. 一般的な仕様
2. X軸のための要素の定義
3. 値系列の仕様
4. レポートを制限する(limit)ために制限(Restrictions)を選択

ステップ2から4は、動的統計を備えたレポートの生成にのみ必要とされます。静的統計のために単に一般情報(point 1)が必要です。

ページを扱う方法に関する情報はヒント・パネル中のアクション・パネル以下にこれらの画面の各々の上で提供されます。

正しくない入力が入力される場合、以前に処理されたユーザ・インターフェースは再び表示れ、それから正しくない入力に関する情報が表示されます。この情報はOTRS通知セクションで見つけることができます。現在の形式が正確に記入された後のみ、次の入力ユーザ・インターフェースが表示されます。

1. 一般的な仕様。それは編集ウイザードの最初のページです(下記の図を参照)。

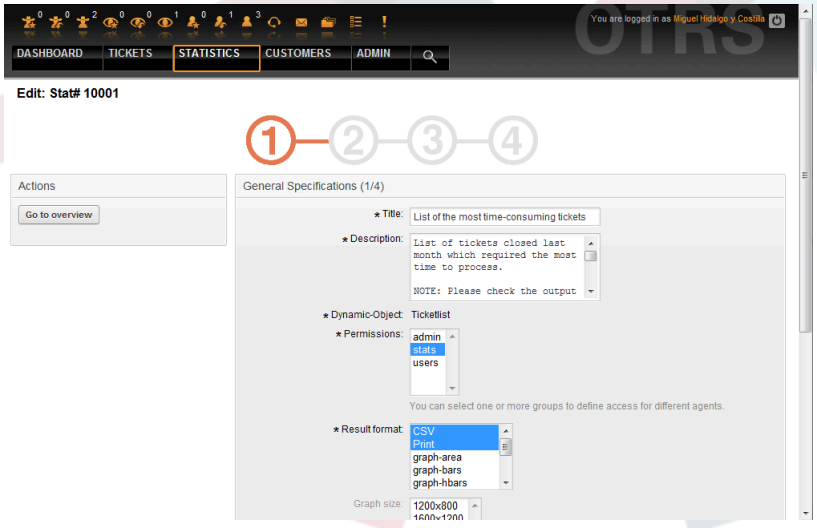


図:レポートの一般的な仕様の編集

図示した画面中で、編集することができる多くの共通の仕様および設定があります:

- タイトル。簡潔なやり方でstatの目的を反映するべきです。
- 摘要。レポート定義、構成パラメータのタイプ等より多くの説明を含む情報。
- 動的オブジェクト。OTRSインストレーションが様々な動的オブジェクトを提供する場合、それらのうちの1つを選ぶことができます。オブジェクトは、特定のモジュールの必要条件を満たします。

- 静的ファイル。通常はこの選択は表示されません。なぜならどのレポートにもまだ割り当てられていない単なる静的ファイルは
  まだ表示されないからです。しかしながら、「静的ファイル」が表示される場合、オプション・フィールドにチェックマークを付けるて、生成モード (動的オブジェクトまたはファイル付の静的オブジェクトで動的) を選択することは重要です。静的ファイルが選択されている場合、入力ユーザ・インターフェイス 2 から 4 は表示されません。静的ファイルはすべての必須の構成設定を含んでいないためです。

- 許可設定。グループ (担当者も) の制限を緩和して下さい。その担当者が事前定義のレポートを後で閲覧し生成できるように。このように様々なレポートを、それらを必要とする異なる部署やワークグループに割り付けることができます。

  例1:「統計」グループが選択されました。少なくとも「統計」グループのためのro権を持っているすべてのユーザは、そのレポートを見ることができます。この権限はデフォルトで利用可能です。

  例2:「販売」という名のグループが選択されました。　「販売」グループのためのro権を持ったユーザは、みなビュー・モードでstatを見て、それを生成することができます。しかしながら、そのレポートは他のユーザよっとビューできません。

- フォーマット。statの出力フォーマット:構成に依存しますが、ひとつのまたは複数の次のフォーマットを選択可能です。

  - CSV。
  - 印刷。
  - グラフ・線
  - グラフ・棒
  - グラフ・横棒
  - グラフ・点
  - グラフ・点。線
  - グラフ・エリア
  - グラフ・パイ

- グラフ化。ピクセル中の図表サイズを選択します。グラフ式の出力フォーマットが「フォーマット」の下で選ばれている場合のみ、この選択は必要です。一般に使用することができるグラフィックのサイズはすべてSysConfigにOTRS管理者によって定義されています。レポートを構成する場合、担当者はあらかじめ適切なフォーマットをすべて選択することができます。

- 行の合計。レポートが列(複数のセルが各行の合計を含んでいる)によって修正されるかどうかを示します。

- 列の合計。レポートが行(複数のセルが各列の合計を含んでいる)によって修正されるかどうかを示します。

- キャッシュ。生成されたレポートがファイルシステムでキャッシュとして保持されるべきかどうかを特定します。レポートが再び呼び出される場合、これは計算能力と時間を節約します。しかし、レポートの内容に変化がない場合、節約になるわけではありません。

  レポートが時間指示値を含んでいない場合、あるいは時間指示値が将来を指す場合、自動的にキャッシングは防止されます。

キャッシュされたレポートが編集される場合、キャッシュされたデータはすべて削除されます。

- 有効。どんな理由でも事前構成されたレポートを一時的に実行してはならない場合、これは「無効」にセットすることができます。その後、右側パネル最下部の「スタート」ボタンは、もはや表示されません。そのレポートはもはや生成することができません。

2. X軸のための要素の定義。それは、X軸の描写に使用された要素、あるいはX軸に適用されたカラム名の(テーブルが使用される場合)構成です(図を参照)。

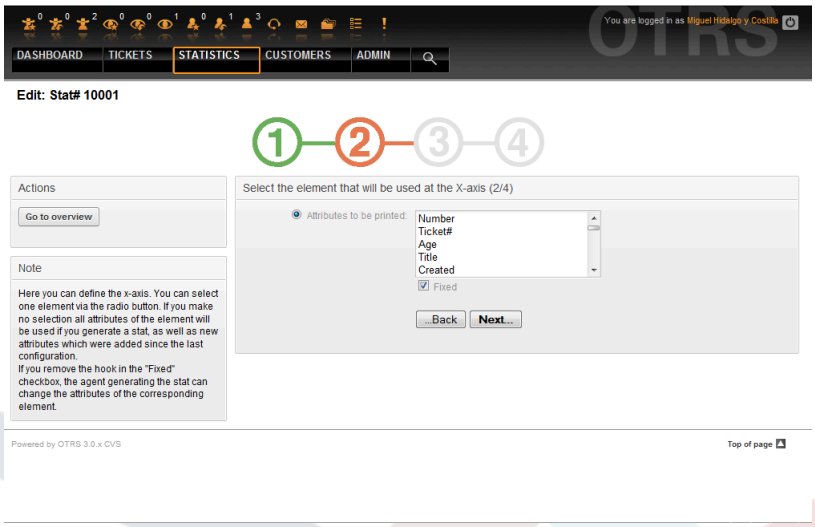


図: X軸のための要素の定義

第一に、要素はオプション・フィールドを使用して選択されています。その後、要素の2つ以上の属性が選択されなければいけません。属性が選択されていない場合、属性はすべてレポートの構成の後に加えられたものを含めて使用されます。

「固定の」セッティングが無効になる場合、レポートを生成する担当者は「ビュー」ユーザ・インターフェース中のそれぞれの要素の属性を変更することができます。

期間と規模を述べなければならないとともに、時間的要因は異なります。要素のタイプおよび数は使用された動的オブジェクトに起因し、それに依存して変わります。

入力がすべて正確な場合、「次」ボタンは「値系列」フォームに向かいます。さらに、初めのセクションの編集に戻ることも可能です。

3. 値系列の仕様

レポート構成の3番目のステップで、値系列は定義されます(下記の図を参照)。それらは、後で表のビュー内で個別のグラフあるいは様々なシリーズを形成するでしょう。

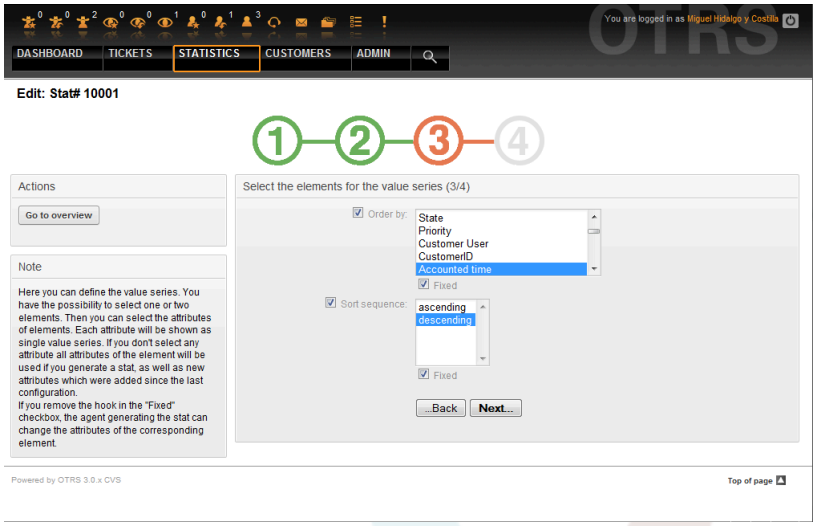


図:値系列の定義

要素が選択されていれば、選ばれた属性はそれぞれ値系列に対応するでしょう(下記の例19-1を参照)。

例4.19 値系列の定義 - 1つの要素

要素キュー:

- 値系列 1 = 未対応
- 値系列 2 = ジャンク
- ....

2つの要素が選択されている場合、最初の要素の選択された属性はそれぞれ、値系列を形成する第2の要素の属性と結合します(下記の例19-2を参照)。

例4.20 値系列の定義 - 2つの要素

要素1 キュー、要素2 ステータス:

- Value chain 1 = 未対応 - 未完了
- Value series 2 = 未対応 - 完了 (成功)
- Value series 3 = ジャンク - 未完了
- Value series 4 = ジャンク - 完了 (成功)

3つ以上の要素の選択は許可されません。

さらに、同じ条件は、「X軸」選択に関してと同様に、属性および「固定の」チェックボックスの選択に当てはまります:

- 要素の属性が選択されていない場合、属性はすべてレポートの構成の後に加えられたものを含めて使用されます。
- 「固定の」セッティングが無効の場合、レポートを生成する担当者はそれぞれの要素の属性を変更することができます。

4. レポートへの制限の設定。これは構成の4番目と最終ステップです(下記の図を参照)。制限は、その結果を選択された基準に制限する役目をします。多くの場合では、制限は全くセット・アップされないかもしれません。

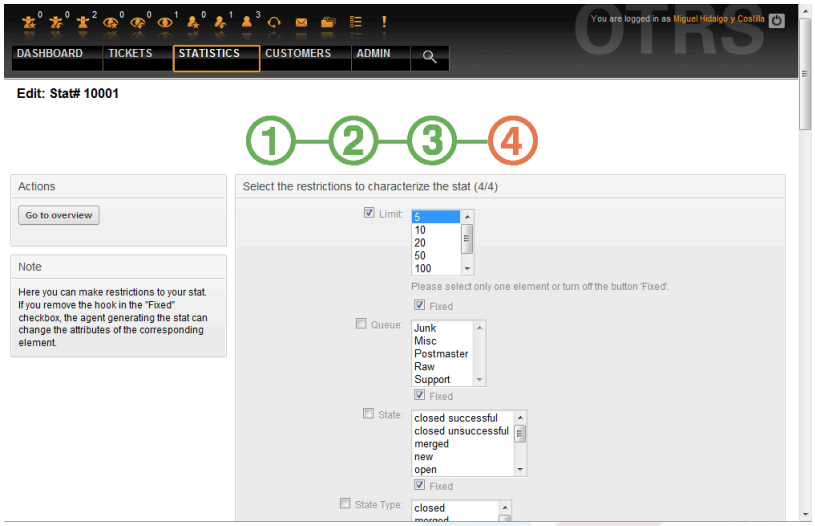


図:制限の定義

制限がすべてセット・アップされた後、レポートの構成は「終了」ボタンを押すことにより完成します。

9.1.4. インポート

インポート・ユーザ・インターフェース(下記の図を参照)はナビゲーション・バー、リンク「統計」から選ぶことによりアクセスすることができ、「インポート。」(その後)一方、概要画面上でインポート・ボタンを押すことは同じ結果となります。レポートへの「rw」権は必須です。

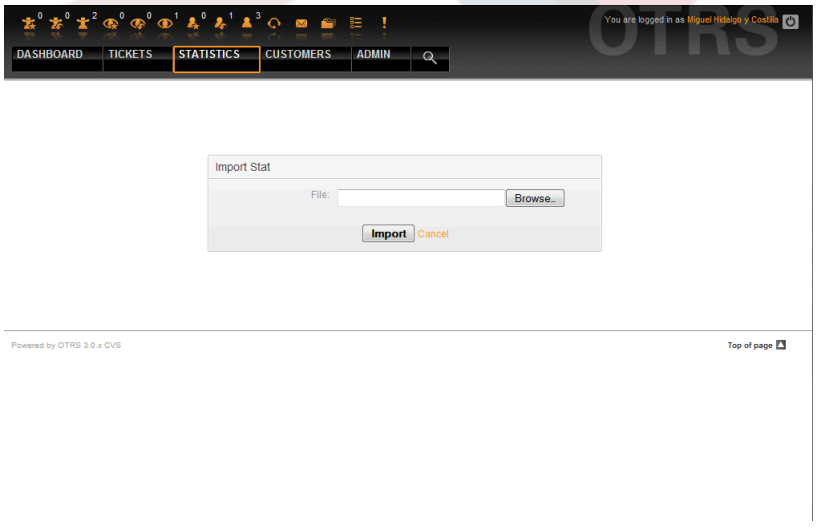


図:インポート・ユーザ・インターフェース

非常で機能的なモジュールのエクスポート機能と結合した時、レポートのインポートはより円滑になります。統計は、テスト・システムに便利なように作成しテストし、次に、プロダクション・システムへインポートすることができます。

インポートは容易なファイル・アップロードによってもたらされます。インポートレポートの「ビュー」ユーザ・インターフェースは自動的に後で開かれます。

9.2. OTRS管理者による統計モジュールの管理

このセクションでは、統計モジュールを取り扱うOTRS管理者のタスクおよび責任に関する情報を提供します。

9.2.1. 許可設定、グループ及びキュー

統計モジュールがインストールされている場合、新規キューおよび(または)グループは作成されません。

モジュール登録ではデフォルト設定で、すべての担当者に対して統計モジュールに「統計」グループ許可アクセスを与えます。

許可設定による権限:

- *rw*。統計とレポートを構成可能にします。
- *ro*。あらかじめ形成された統計およびレポートを構成可能にします。

OTRS管理者は、以下のふたつを決定します。ひとつは、事前構成されたレポートを生成する許可を持つ担当者が、「統計」グループ中でro権を割り当てられるか、もうひとつは、それぞれのグループがSysConfigの中のモジュール登録で追加されるかどうかです。

9.2.2. SysConfig

The SysConfig groups Framework:Core::Stats, Framework:Core::Stats::Graph and Framework:Frontend::Agent::Stats contain all configuration parameters for the basic set-up of the statistics module. Moreover, the configuration parameter $Self->{'Frontend::Module'}->{'AgentStats'} controls the arrangement and registration of the modules and icons within the statistics module.

9.3. システム管理者による統計モジュールの管理

一般に、システム管理者は、統計モジュールのオペレーション、構成およびメンテナンスに必要とされません。しかしながら、この点に関してシステム管理者に若干の背景情報は与えられます。

> **注記**
>
> ファイル・パスは、OTRSホームディレクトリー(ほとんどの場合 `/opt/otrs`)のサブディレクトリを指します。

9.3.1. データベース・テーブル

レポート構成はすべてXMLの中でインプリメントされ処理され、したがって、データベース・テーブル「xml_storage」に格納されます。その内容がxmlフォーマットで示される他のモジュールは同様にこのテーブルを使用します。

9.3.2. すべてのファイルのリスト

以下は、統計モジュールが正確に動作するためにのに必須のファイルです:

- `Kernel/System/Stats.pm`
- `Kernel/Modules/AgentStats.pm`
- `Kernel/System/CSV.pm`
- `Kernel/Output/HTML/Standard/AgentStatsOverview.dtl`
- `Kernel/Output/HTML/Standard/AgentStatsDelete.dtl`
- `Kernel/Output/HTML/Standard/AgentStatsEditSpecification.dtl`

- Kernel/Output/HTML/Standard/AgentStatsEditRestrictions.dtl
- Kernel/Output/HTML/Standard/AgentStatsEditXaxis.dtl
- Kernel/Output/HTML/Standard/AgentStatsEditValueSeries.dtl
- Kernel/Output/HTML/Standard/AgentStatsImport.dtl
- Kernel/Output/HTML/Standard/AgentStatsPrint.dtl
- Kernel/Output/HTML/Standard/AgentStatsView.dtl
- Kernel/System/Stats/Dynamic/Ticket.pm
- bin/otrs.GenerateStats.pl

9.3.3. キャッシング

統計の結果がキャッシュされるかされないかは構成でセットアップされます。キャッシュ・レポート結果は、OTRSインストレーション(ほとんどの場合/opt/otrs/var/tmp)のvar/tmp directoryでファイルとして格納されます。

キャッシュ・統計は、「Stat」接頭辞によって認識することができます。

データが失われるにしても、大きな被害を引き起こしません。次回レポートが呼び出される時には、統計モジュールはファイルを見つけることなく、新規レポートを生成するでしょう。もちろん、これは恐らく実行にもう少し長くかかるでしょう。

9.3.4. otrs.GenerateStats.pl

このファイルはbin directoryに保存されます。それは、コマンドライン中のレポートの生成に役立ちます。

例として、次のスクリプトでコマンドライン呼び出しを参照してください。

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

スクリプト：コマンドラインからのレポートの生成

統計構成「Stat# 10004」からのレポートは/output/dir ディレクトリー中でcsvとして生成され保存されます。

生成されたレポートも、Eメールとして送ることができます。より多くの情報は、スクリプト中のコマンドで下に呼び出すことができます。

```
bin> perl otrs.GenerateStats.pl --help
```

スクリプト: *otrs.GenerateStats.pl*ファイルに関する情報の入手

9.3.5. 自動統計生成 - Cronジョブ

統計モジュールに便利なGUIがあるため、通常はコマンドラインによってレポートを手動で生成しても意味をなしません。しかしながら、Cronジョブと結合した時、手動によるレポートの生成は意味をなします。

次のシナリオを想像してください:毎月の1日目に、部長は先月のレポートを受け取りたい。cronジョブとコマンドラインの呼び出しを組み合わせることによって、レポートは、Eメールでそれらに自動的に送ることができます。

9.3.6. 静的な統計

統計モジュールは、静的統計の生成に役立ちます。全ての静的統計はそのコンテンツが正確に定義されているファイルに存在します。

このように非常に複雑な統計が生成されます。そのデメリットはそれらが特に柔軟ではないことです。

ファイルはディレクトリKernel/System/Stats/Static/　に保存されます。

9.3.7. 古い静的な統計の使用

先のOTRSバージョン1.3および2.0は、既に統計/レポートの生成を促進しました。顧客の必要条件を満たすために特に開発されているOTRSバージョン1.3および2.0のための様々なレポートは、最近のOTRSバージョンの中でまた使用することができます。

ただ、ファイルは、Kernel/System/Stats/ のパスからKernel/System/Stats/Static/ のパスへ移動されなければいけません。さらに、「::Static」によってそれぞれのスクリプトのパッケージ名を修正しなければなりません。

次の例は、最初のパスがどのように修正されるか示します。

```
package Kernel::System::Stats::AccountedTime;
```

```
package Kernel::System::Stats::Static::AccountedTime;
```

9.3.8. Default stats (デフォルト統計)

「わかり切ったことを最初からやり直すことは必ずしも必要だとは限りません...」

統計モジュールは様々なデフォルトレポートを提供します。すべてのOTRSユーザに興味深いレポートは、統計モジュール・パッケージのデフォルトレポートセットに今後加えられるでしょう。デフォルトレポートは、scripts/test/sample/ ディレクトリー中の統計モジュールxmlフォーマットで保存されます。

10. 動的フィールド

10.1. 導入

動的フィールドは、チケットか記事上に格納された情報を拡張するために作られて、OTRSの中の特別の種類のフィールドです。これらのフィールドはシステムに固定されません。 そして、それらは特定の画面にのみ現われる場合があります。それらが必須であれ、そうでなかれ、それらの画面上の表現はそれらの作成時間に定義されたフィールドタイプに依存します。その作成時間はフィールドによって保持されるデータよります。例えば、テキスト、日付、アイテムの選択を保持するフィールドがあります。

動的フィールドは、TicketFreeText　TicketFreeKey　TicketFreeTime、ArticleFreeTextおよびArticleFreeKeyフィールドの発展形です。それらは以前もそしてOTRS　3.0の中でも一般に使用されます。これらの「自由フィールド」には以下の制限がありました。1.)　16の(テキストまたはdropdown)フィールドまで定義可能。2.) 1チケットにつき6回のフィールドまで定義可能。3.) 1記事につき3フィールド(テキストまたはdropdown)まで定義可能。

今、動的フィールドで、チケットまたは記事についてフィールドの数の中の制限は撤去される、チケットか記事のいずれかのために希望する数だけ動的フィールドとして作成することができます。また、それを越えて、動的フィールドの背後のフレームワークは、単なるチケットや記事だけはなく、他のオブジェクト用カスタム・フィールドを扱う準備ができています。

動的フィールドを扱うこの新規フレームワークは、モジュール的アプローチを使用して、構成されている。ここでの各種類の動的フィールドは、フレームワーク用のプラグインモジュールと見なすことができます。このことは、公開されているOTRSモジュール、OTRSアドオン、OTRSカスタム開発および他のカスタム開発によって容易に動的フィールドの種類を拡張することができることを意味します。

次の動的フィールド・タイプがこのリリースで含まれている:

- テキスト(テキストの1行のライン)
- テキスト領域(テキストの多数の行)
- チェックボックス
- ドロップダウン(単一の選択、複数の値)
- マルチ選択(多項式選択の多数の値)
- 日付
- Date / Time

10.2. 構成

デフォルトで、OTRSのクリーン・インストールは動的フィールドを含んでいません。チケットまたは記事の中でそのようなフィールドを使用することを計画すれば、動的フィールドを作る必要があります。

動的フィールドの構成はふたつの部分に分かれています。ひとつは新しい動的フィールドを追加するため。もうひとつは、既存の動的フィールドを管理するためです。（既存のものは「動的フィールド」リンクにある「管理者」パネルへとあなたが案内する必要があります。）画面の中で、必須として表示するため、あるいは動的フィールドを非表示にするため、あなたは「SysConfig」画面の中でOTRS設定を変更する必要があります。

10.2.1. 動的フィールドの追加

ナビゲーション・バーにある「管理者」ボタンをクリックして、次に、画面の底部中央付近にある「チケット設定」ボックスの内部の「動的フィールド」リンクをクリックしてください。概要が以下のように表示する動的フィールド:

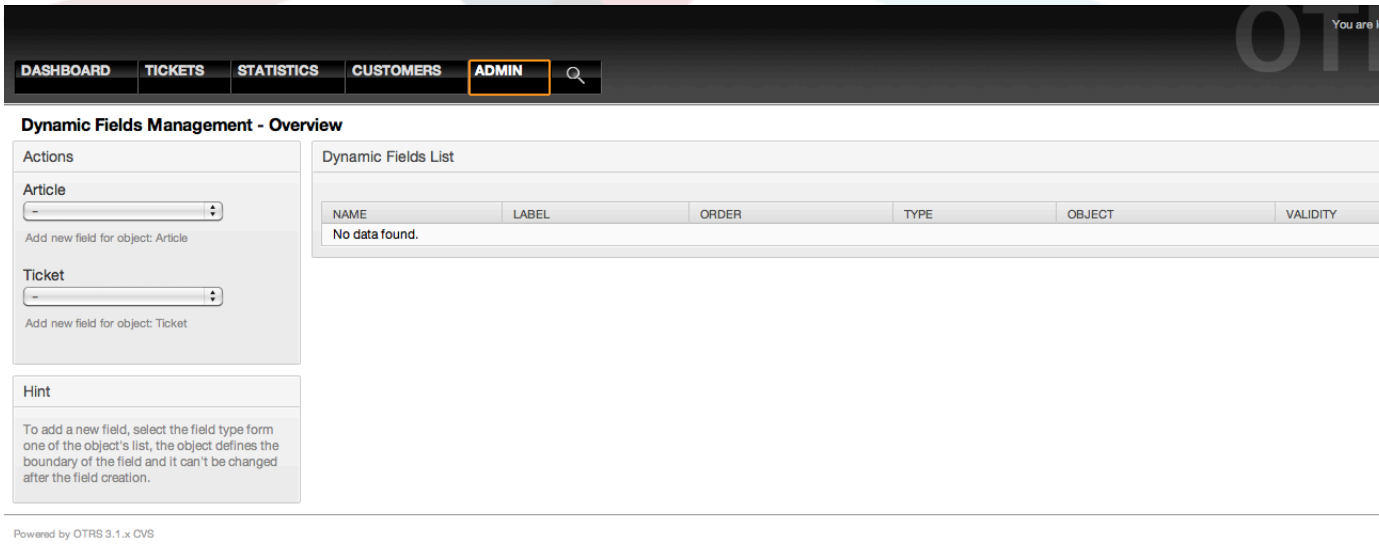


図:動的フィールド概要画面、空

作られてた全ての動的フィールドをリストへ、あなたがさらに動的フィールドを追加すると、この画面が変化することにご留意下さい。インストレーションがOTRSのより古いバージョンから更新された場合、この画面には既にいくつかのフィールドがあるかもしれません。

画面の左のサイド・バーのアクションは、2つの可能性について表しています: それは記事およびチケットで、各々は動的フィールドの自身のdropdown選択を持っています。

注記

OTRSパッケージのインストレーションはより多くのオブジェクトをアクション・サイド・バーに追加しました。

動的フィールドを作る基本手順は次のとおりです:

- アクション・サイド・バーで希望するの動的フィールドオブジェクトdropdownをクリックしてください。
- リストから追加したい動的フィールドタイプをクリックしてください。
- 構成を記入してください。
- 保存してください。

動的フィールドのための構成ダイアログはふたつの部分に分かれます。上部のセクションはすべてのフィールド中に一般的です。また、下方部分は別のタイプへの1つのタイプの動的フィールドとは異なるかもしれません。

一般的な動的フィールドセッティング:

- 名前:必須で、ユニークで、文字および数字だけが許可されます。

  これは、1つの画面の中で例えばフィールドを表示または非表示するために使用するような、フィールドの内部名前である。フィールド名のどんな修正する(推奨しませんが)場合でも、フィールドがreferenciated (参照) される「SysConfig」設定の手動のアップデートが必要です。

- ラベル：必須

  これは画面に表示されるフィールド名である。それは翻訳を支援します。

  ### 注記

  ラベルの翻訳を言語変換ファイルに手動で加えなければならない。

- フィールド順：必須

  相対的な順序を定義します。画面で表示されるフィールドの相対的な順序を定義します。デフォルトでは各々の新規フィールドは最後尾に位置します。この設定の変更は他に作られた動的フィールドの順序に影響します。このセッティングの変化は、他の作られた動的フィールドの順序に影響するでしょう。

- 有効性:必須

  如何に表示するように構成しても、無効の動的フィールドはどの画面にも表示されません。

- フィールド・タイプ:必須。読み込み専用。

  現在の選択されたフィールド・タイプを示します。

- オブジェクト・タイプ:必須。読み込み専用

  フィールドの範囲を示します。

  ### 注記

  例では、特定のフィールド・タイプの設定を各々説明するために、少数のフィールドが追加されます。これらの新規フィールドは後のセクションの中でreferenciated (参照) されるでしょう。

  次の例については、記事オブジェクトのための動的フィールドを作る必要があれば、チケット・オブジェクトのためにすべての動的フィールドが作られます。単に記事ドロップダウン・リストからフィールドを選択して下さい。

表**4.6** 次のフィールドがシステムへ追加されます。:

Name	ラベル	タイプ
Field1	My Field 1	テキスト
Field2	My Field 2	Textarea
Field3	My Field 3	チェックボックス
Field4	My Field 4	ドロップダウン
Field5	My Field 5	Multiselect
Field6	My Field 5	日付
Field7	My Field 6	Date / Time

10.2.2. テキスト動的フィールドの構成

テキスト動的フィールドは単一行の文字列を格納するために使用されます。

テキスト動的フィールド設定:

- デフォルト値:オプション。

  これは、デフォルトでは編集画面上に表示される値です。(新規電話チケットまたはチケットが作成されるような画面)

- リンクの表示：オプション。

  セットされたら、フィールド値は、表示画面用に (チケット・ズームや概要のように) クリック可能なリンクに変換されます。

  例えば、「リンクを表示」を「http://www.otrs.com」にセットするなら、値を入力してクリックすれば、ブラウザでOTRSウェブページが開きます。

  ### 注記

  セット・リンク値中の　$LQData{「NameX」}を使うなら、NameX　があるところに、フィールドの名前がリンク参照の一部としてフィールド値を加えるでしょう。

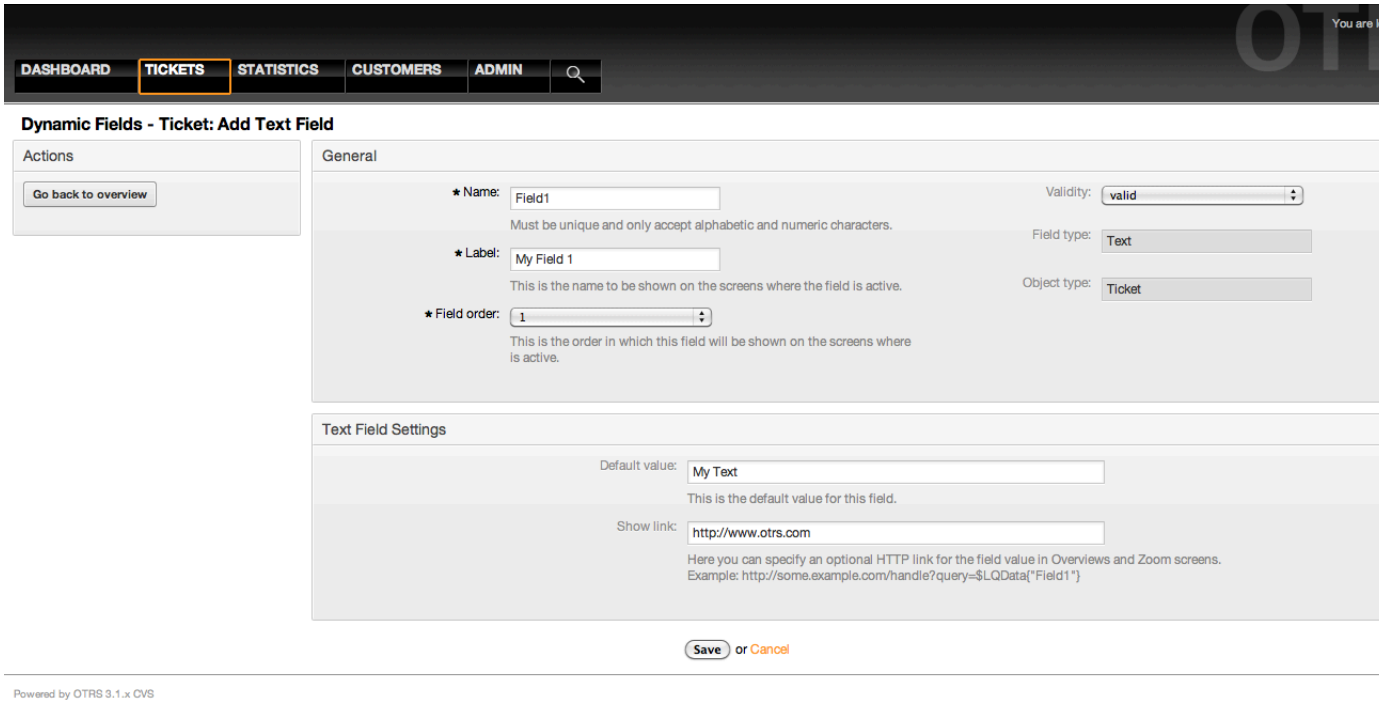


図:動的フィールド・テキストの構成ダイアログ

10.2.3. テキスト領域動的フィールド構成

テキスト領域の動的フィールドは複数行の文字列を格納するために使用されます。

テキストエリア動的フィールドの設定:

- 列の数:オプション、整数。

  編集画面(新規電話チケットあるいはチケット作成のように)にフィールドの高さを定義するために使用されます。

- 列の数：オプション、整数。

  この値は、編集画面でフィールドの幅を定義するために使用されます。

- デフォルト値:オプション。

  これは、編集画面中のデフォルトによって示される値です。(複数行テキストも可)

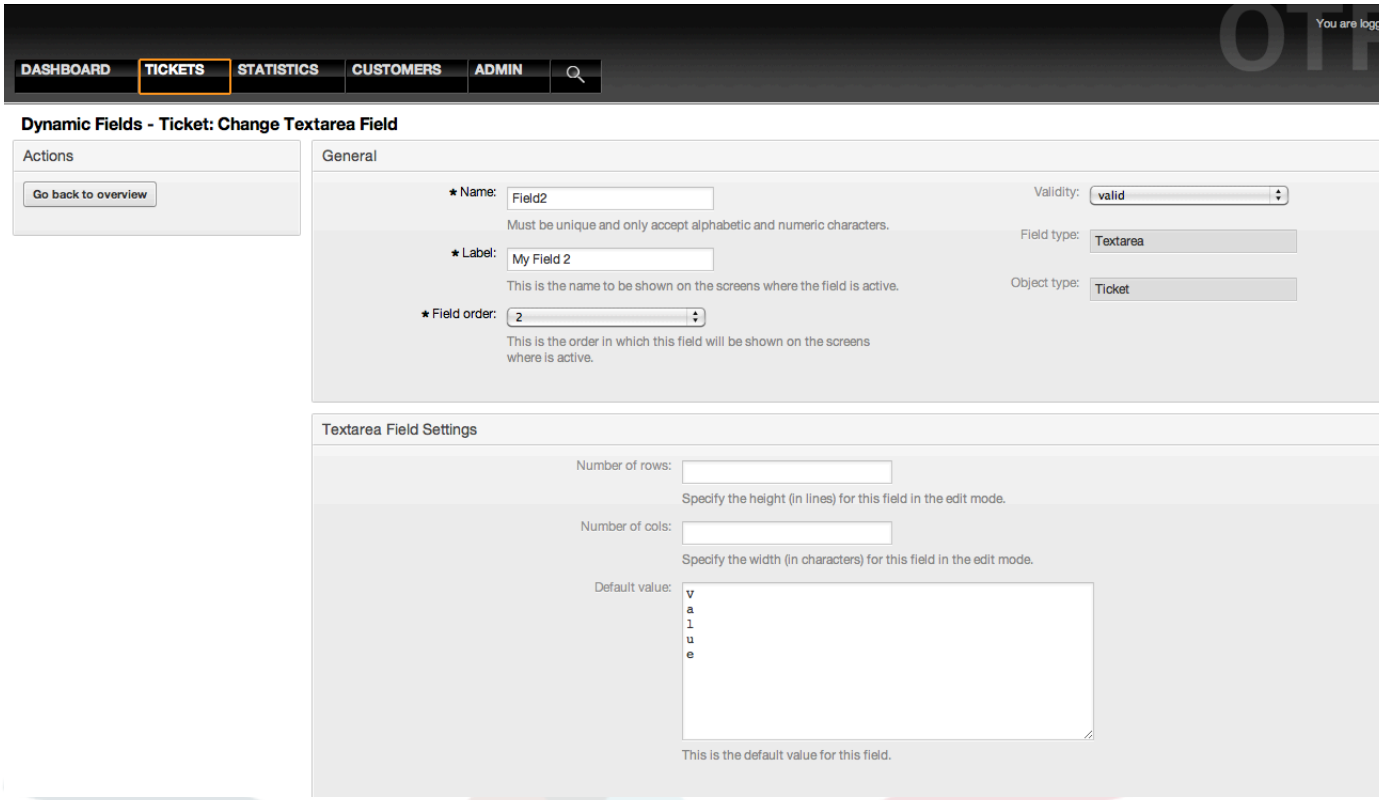


図:動的フィールドテキストエリアの構成ダイアログ

10.2.4. チェックボックス動的フィールド構成

チェックボックス動的フィールドは、真偽値を格納するために使用されます。チックボックス内のチェックの有無で表現されます。

チェックボックス動的フィールドの設定:

- デフォルト値:必須。

  これは、編集画面上デフォルトによって示される値です。(新規電話チケットあるいはチケット作成のように)このフィールド用のデフォルト値は、チェックの有無をすることができる閉じた選択です。

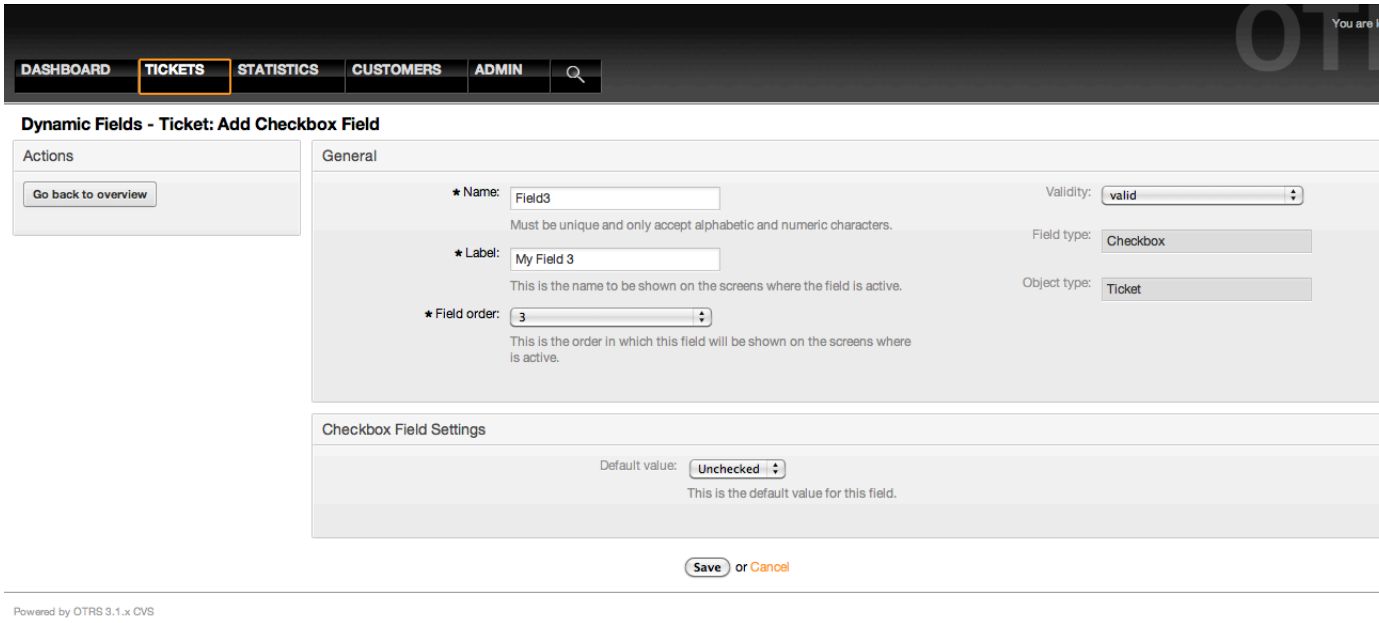


図:動的フィールド・チェックボックスの構成ダイアログ

10.2.5. ドロップダウン動的フィールドの構成

ドロップダウン動的フィールドは単一の値を格納するために閉じたリストから使用されます。

ドロップダウン動的フィールドの設定:

- 可能な値：必須。

  例えば、「リンクを表示」が「http://www.otrs.com」にセットするなら、値を入力してクリックすれば、ブラウザでOTRSウェブページが開きます。

- デフォルト値:オプション。

  この値はデフォルトで編集画面に表示されます。(新規電話チケットあるいはチケット作成のように)このフィールドのデフォルト値は、可能な値によって定義された閉じた選択です。

- 空の値の追加：必須、(はい/いいえ)

  このオプションが活性化されるなら、特別な値が、可能な値のリスト中で「-」を表示されて定義されます。この特別な値は内部的には空です。

- 翻訳可能な値:必須、(はい/いいえ)

  この設定は、このフィールドが翻訳可能な値であることの指標として使われます。単に翻訳される表示上の値であり、内部的な値に影響を与えるものではありません。翻訳値は手動で言語ファイルに追加する必要があります。

- リンクの表示：オプション。

  セットされたら、フィールド値は
  表示画面(ズームや概要のように)用にクリック可能ならHTPリンクに変換されます。

例えば、もし「表示」リンクが「 http://www.otrs.com 」にセットされたら、フィールド値をクリックすればブラウザでOTRSのウェブページが開きます。

注記

セット・リンク値中で $LQData{「NameX」} が使われた場合、NameX がフィールドの名前なら、リンク参照の一部としてフィールド値を加えるでしょう。

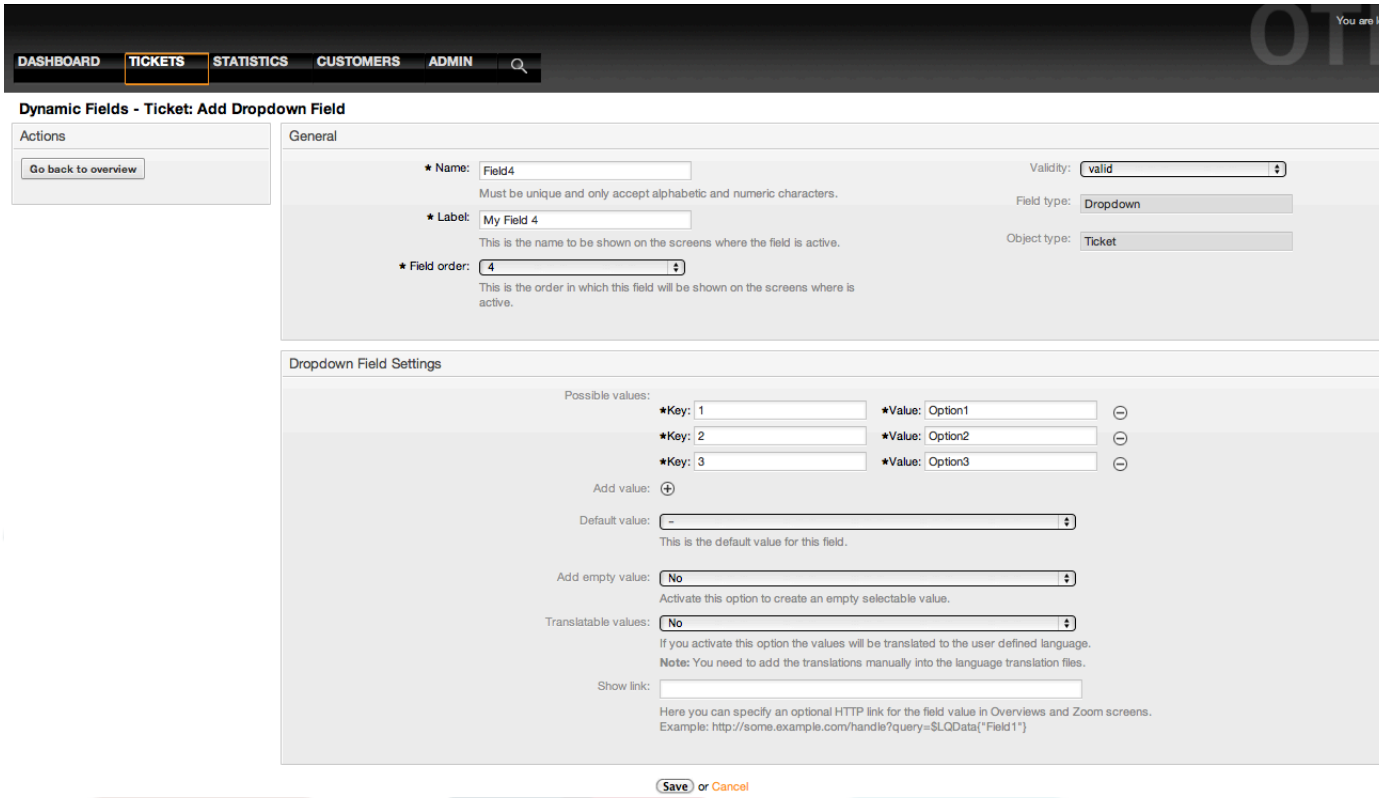


図:動的フィールド・ドロップダウンの構成ダイアログ

10.2.6. 多重選択動的フィールドの構成

多重選択動的フィールドは多数の値を格納するために閉じたリストから使用されます。

多重選択動的フィールドの設定:

- 可能な値：必須。

  フォームを選択するための値のリスト。
  追加のリスト項目を追加するなら、キー（内部値）と値（表示値）を指定する必要があります。

- デフォルト値:オプション。

  これは、編集画面上にデフォルトで示される値です(新規電話チケットあるいはチケット作成のように)。このフィールドのデフォルト値は、可能な値によって定義されるような閉じた選択です。

- 空の値の追加：必須、(はい/いいえ)

  このオプションが活性化される場合、特別な値が、可能な値のリスト中で「-」を表示されて定義されます。この特別な値は内部的には空です。

- 翻訳可能な値:必須、(はい/いいえ)

この設定は、このフィールドが翻訳可能な値であることの指標として使われます。単に翻訳される表示上の値であり、内部的な値に影響を与えるものではありません。翻訳値は手動で言語ファイルに追加する必要があります。

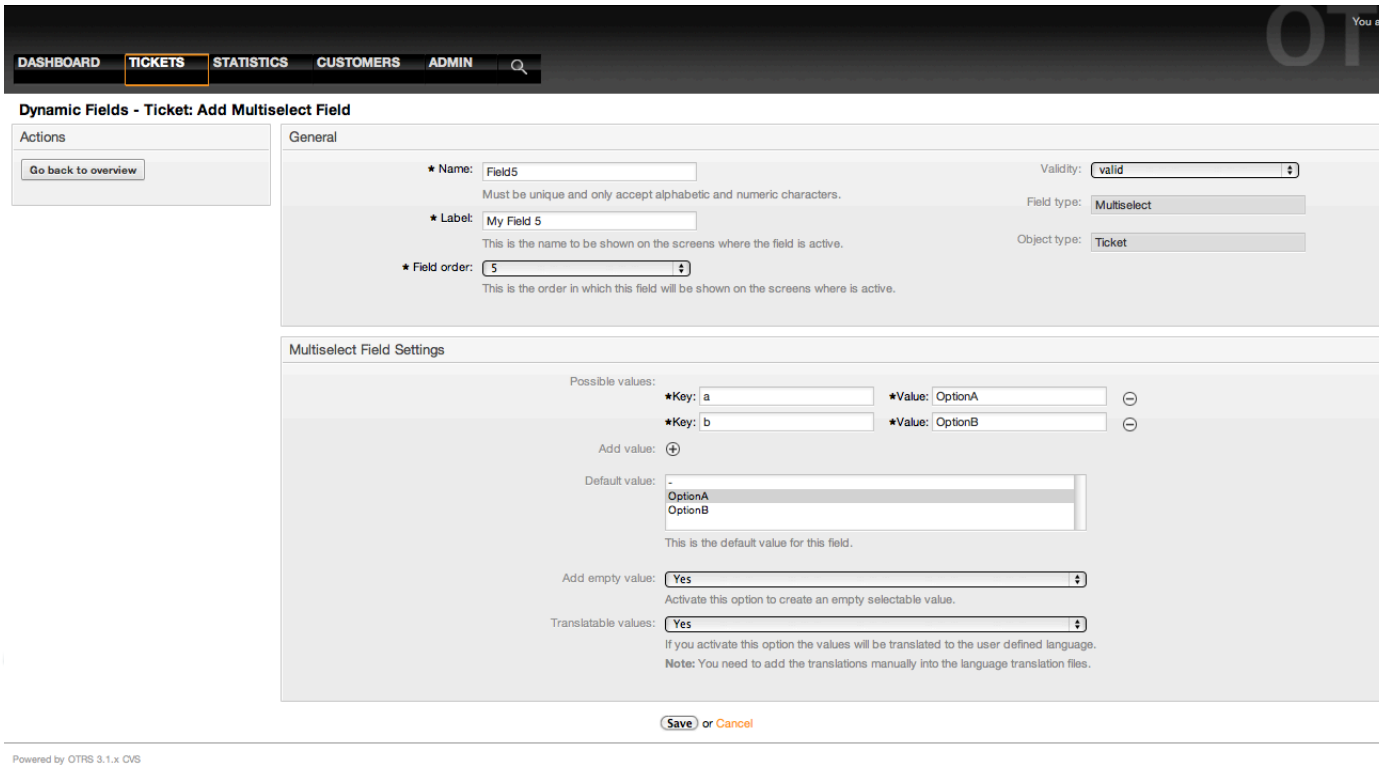


図:動的フィールド 複数選択の構成ダイアログ

10.2.7. 日付動的フィールド設定

日付動的フィールドは日付値(日、月および年)を格納するために使用されます。

日付動的フィールド設定:

- デフォルトの時差：オプション、整数

  現在の日付と選択された日付間の秒数（正でも負でも）。デフォルトで編集画面に表示されます。（新規電話チケットやチケット作成のように）

- 年の期間を定義：必須（はい/いいえ）

  この設定は、このフィールドが翻訳可能な値であることの指標として使われます。単に翻訳される表示上の値であり、内部的な値に影響を与えるものではありません。翻訳値は手動で言語ファイルに追加する必要があります。

  - 過去の年：オプション、正の整数

    編集画面の年選択で表示する現在の日付から過去の年数を定義します。

  - 将来の年：オプション、正の整数

    編集画面にある年選択で表示するために現在の日付から将来の年数を定義します。

- リンクの表示：オプション。

セットされたら、フィールド値は
　表示画面（ズームや概要のように）用にクリック可能ならHTPリンクに変換されます。

例えば、もし「表示」リンクが「　http://www.otrs.com　」にセットされたら、フィールド値をクリックすればブラウザでOTRSのウェブページが開きます。

注記

セット・リンク値中の　$LQData{「NameX」}を使うなら、NameX　があるところに、フィールドの名前がリンク参照の一部としてフィールド値を加えるでしょう。

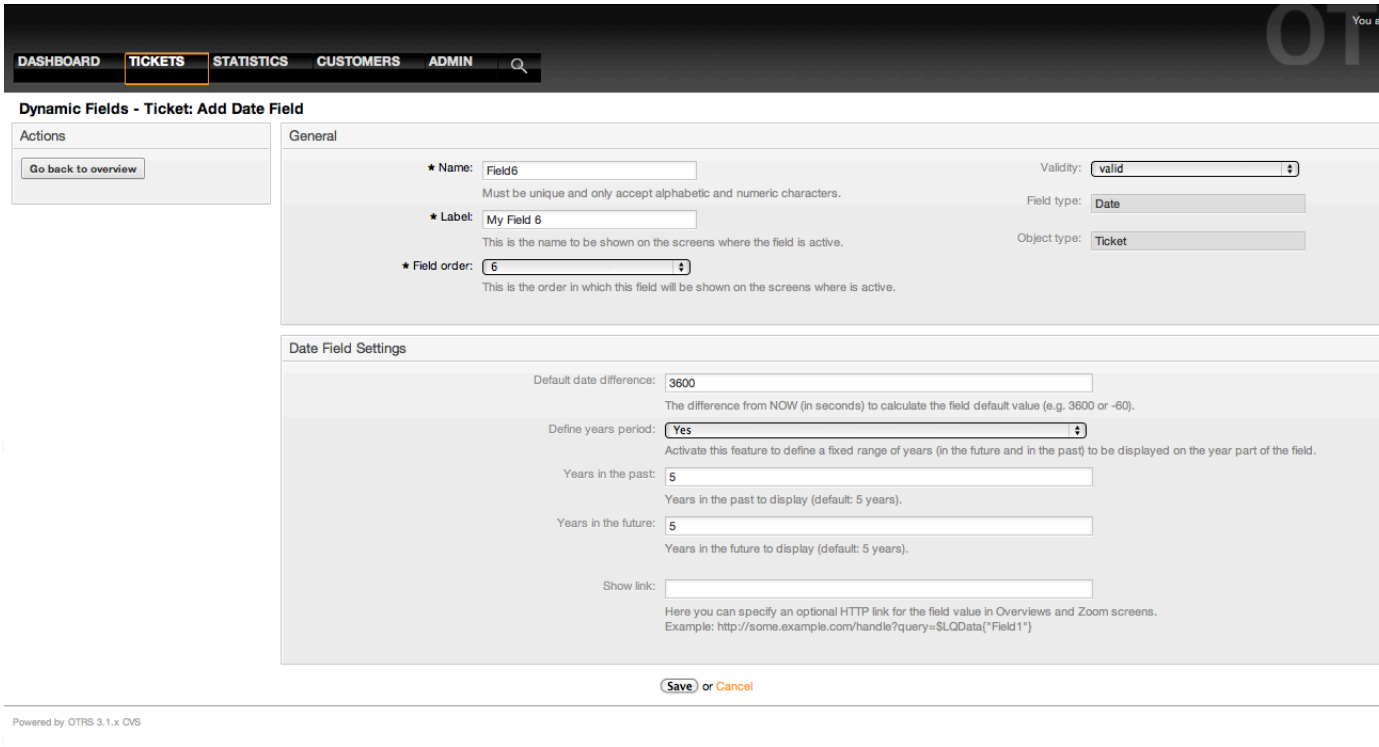


図:動的フィールド 日付/時刻の構成ダイアログ

10.2.8. 日付/時刻　動的フィールドの編集

日付/時刻
　動的フィールドは日付/時刻の値（分、時、日、月と年）を格納するために使用されます。

日付/時刻　動的フィールドの設定：

- デフォルトの時差：オプション、整数

  編集画面でデフォルトで表示される（新規電話チケットまたはチケット作成のように）現在の日付と選択した日付間の秒数（正負可）

- 年の期間を定義：必須（はい/いいえ）

  このフィールドの年設定で、過去に将来（現在日から直近３日）に定義済の年数を設定するために使います。次のオプションが「はい」に設定されて利用可能ならば、：

  - 過去の年：オプション、正の整数

    編集画面の年選択で表示する現在の日付から過去の年数を定義します。

- 将来の年：オプション、正の整数

  編集画面にある年選択で表示するために現在の日付から将来の年数を定義します。

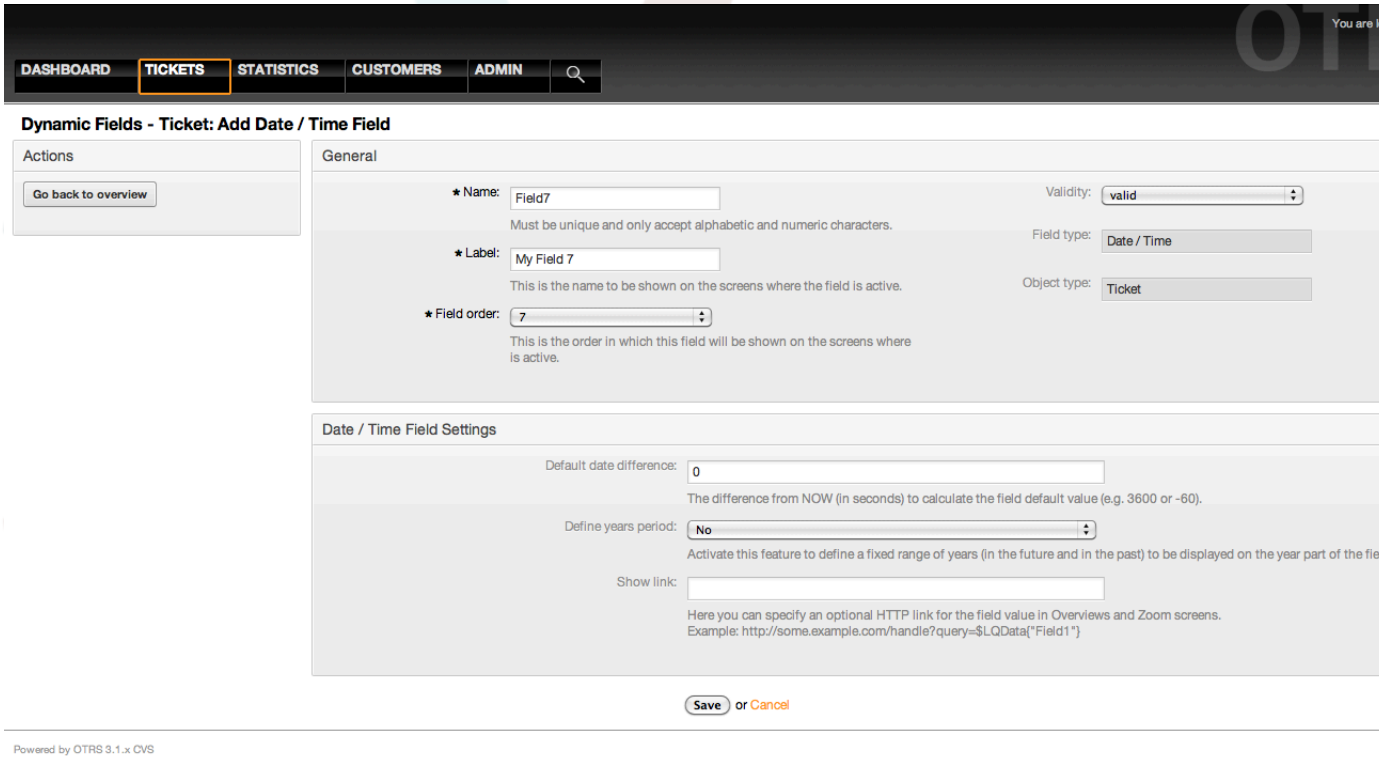
- リンクの表示：オプション。

  セットされたら、フィールド値は
  表示画面（ズームや概要のように）用にクリック可能ならHTPリンクに変換されます。

  例えば、もし「表示」リンクが「 http://www.otrs.com 」にセットされたら、フィールド値をクリックすればブラウザでOTRSのウェブページが開きます。

  ### 注記

  セット・リンク値中の $LQData{「NameX」}を使うなら、NameX があるところに、フィールドの名前がリンク参照の一部としてフィールド値を加えるでしょう。



図：動的フィールド 日付 の構成ダイアログ

10.2.9. 動的フィールドの編集

入力された動的フィールドの概要画面は（前回のサンプルと）同様に見えるはずです。

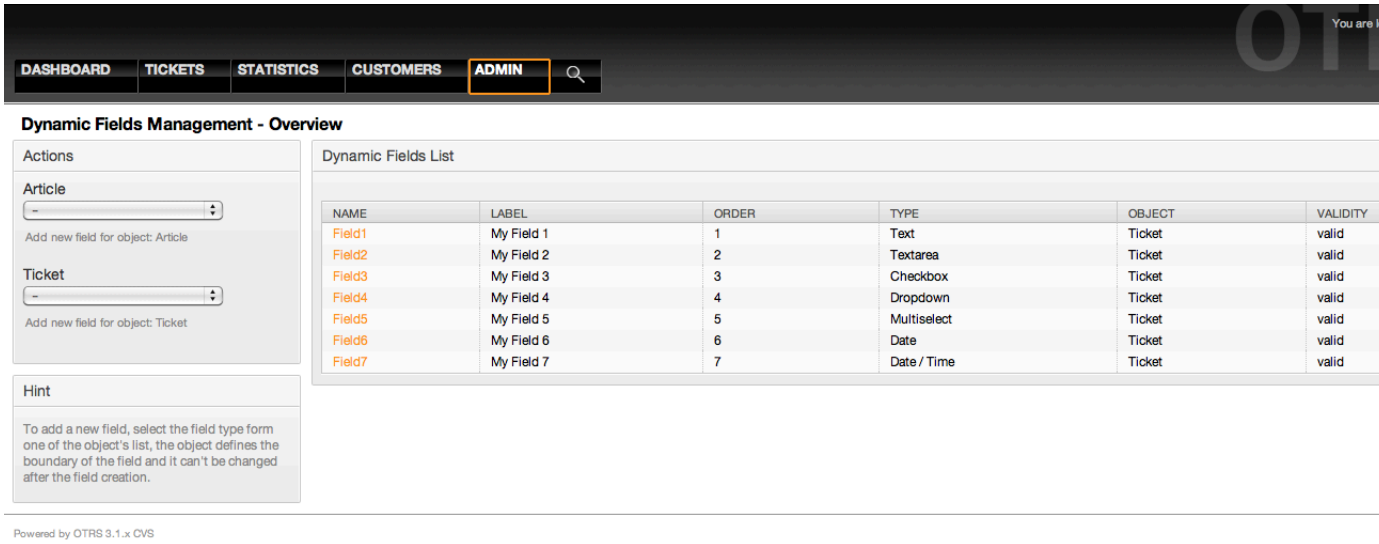


図:サンプルデータで埋められた動的フィールドの概要画面

動的フィールドを変更・編集するために、少なくともひとつのフィールドを定義し、動的フィールドの概要画面から既に追加済フィールドを選択し、その設定を更新しなければいけません。

注記

すべての動的フィールド設定が変更できるわけではありません。フィールド・タイプとオブジェクト・タイプはフィールドの選択から固定し、それらを変更することができません。

ラベルはいつでも変更することができますが、フィールド内部名の変更は推奨されません。もし内部名が変更されるならば、全ての特別なフィールドを参照するSysConfig設定は、ユーザ選択 (定義されていれば) 同様に更新されるべきです。

10.2.10. 画面上での動的フィールドの表示

特別の画面に動的フィールドを表示するために、2つの必須条件があります:

1. 動的フィールドは有効でなければいけません。

2. 動的フィールドは画面の構成の中で1または2にセットされなければいけません。

画面の中で動的フィールドを表示する次のステップに進みます。

- 必ず動的フィールドを有効にセットして下さい。フィールドフォームの有効性を動的フィールド概要画面から確認できます。必要ならフィールドを編集して有効にして下さい。
- 「Sysconfig」を開き、画面左側のアクション・サイドバーにあるドロップダウンリストから「チケット」を選択して下さい。

注記

さらに、既にそれを知っていれば、上記の検索ボックスあるいは「sysconfig」キー中の「動的フィールド」を直接検索することができます。

- 探している画面用サブグループ設定へ行ってそれをクリックします。例えば「Frontend::Agent::Ticket::ViewPhoneNew」。
- "###DynamicField"で終わるセッティングを検索してください。例えば"Ticket::Frontend::AgentTicketPhone###DynamicField"。
- セッティングが空か必須の動的フィールド名がない場合は、新規エントリーを加えるためには「+」ボタンをクリックしてください。例えばキー:Field1、Content:1 .

  設定にすでに動的フィールド名がリストされてることが確かならば、フィールドに表示するためには「1」をセットし、必須として表示するためには「2」をセットします。
- 画面底部にある「アップデート」ボタンをクリックする、及びあなたがフィールドを表示したい画面へ導いて構成を保存して下さい。

10.2.10.1. サンプルの表示

下記は、異なる画面上の動的フィールドを表示するか非表示にするかの「sysconfig」構成例です。

例**4.21** 新規電話チケット画面にあるフィールド 1 を活性化します。

- グループ: チケット
- サブグループ: Frontend::Agent::Ticket::ViewPhoneNew
- 設定: Ticket::Frontend::AgentTicketPhone###DynamicField
- 値:

キー	内容
Field1	1

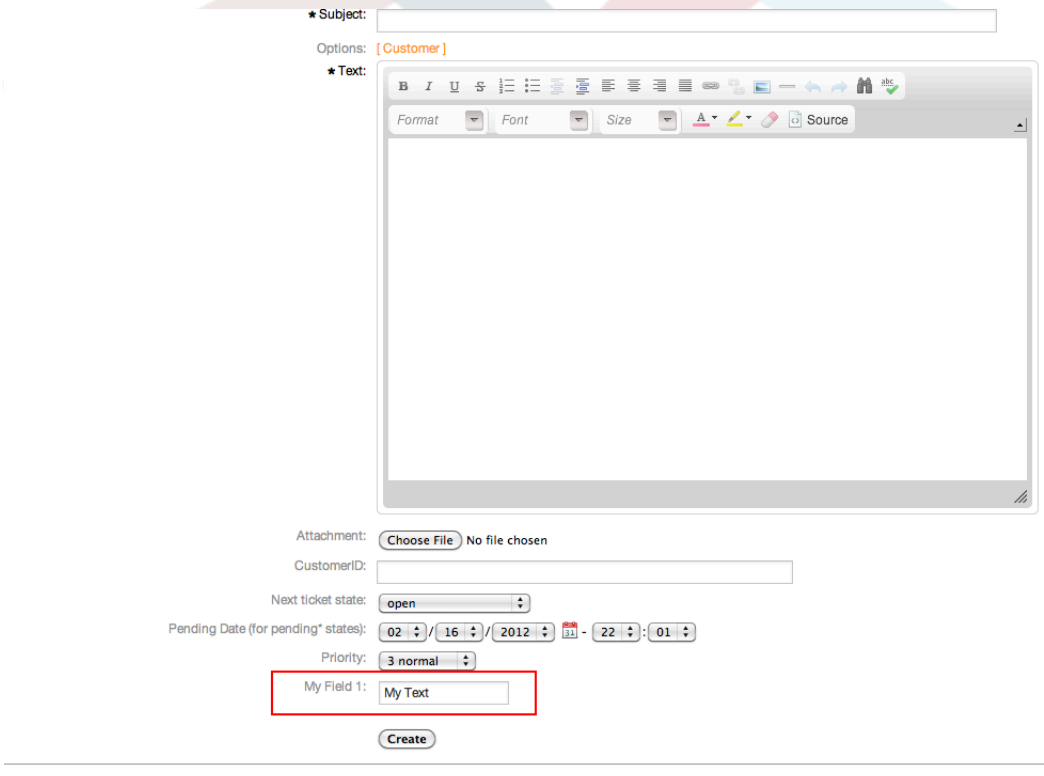
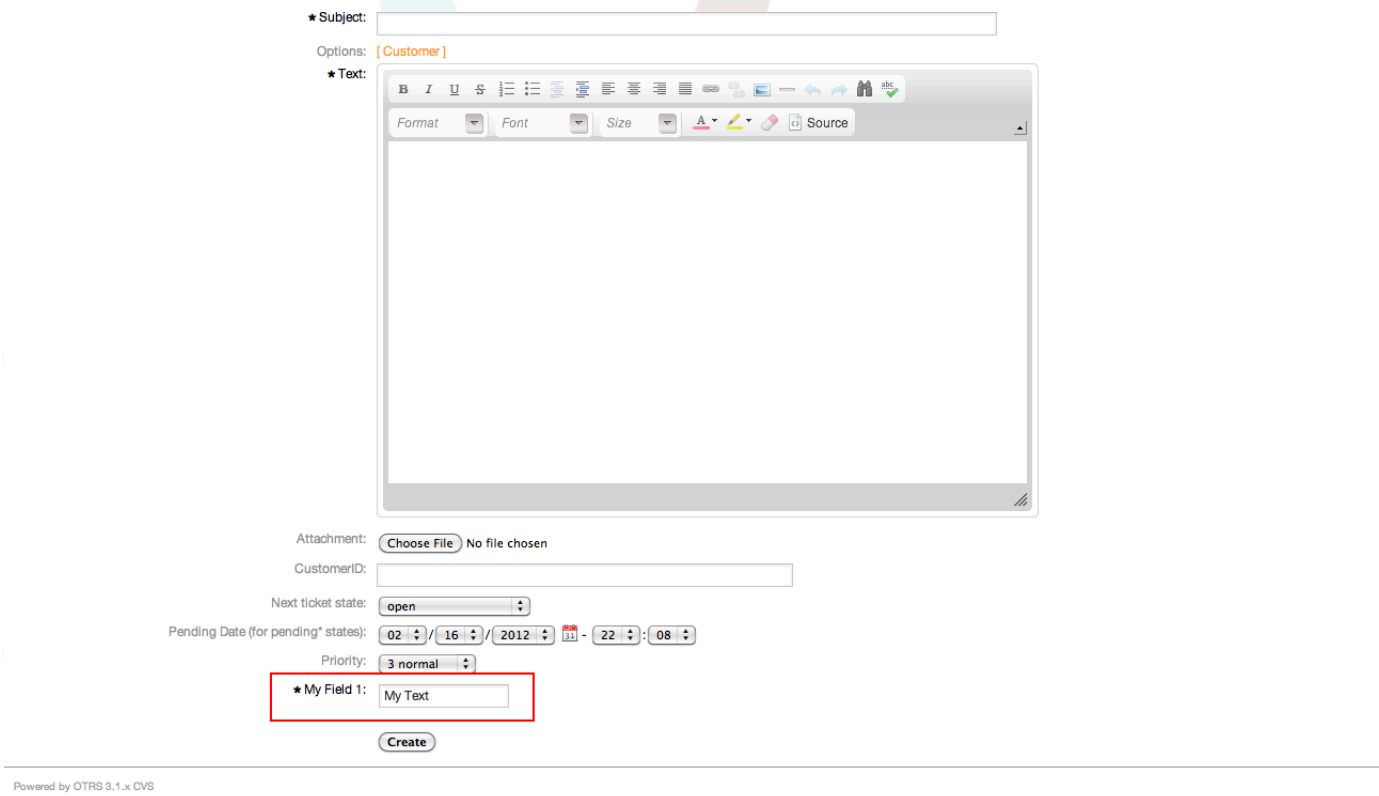


図:新規電話チケット画面中の*Field1*

例4.22 新規電話チケット画面中の必須フィールド1を活性化

- グループ: チケット
- サブグループ: Frontend::Agent::Ticket::ViewPhoneNew
- 設定: Ticket::Frontend::AgentTicketPhone###DynamicField
- 値:

キー	内容
Field1	2



図：新規電話チケット画面中の必須フィールド*1*

例4.23 必須として新規電話チケット画面中のフィールド1を活性化

- グループ: チケット
- サブグループ: Frontend::Agent::Ticket::ViewPhoneNew
- 設定: Ticket::Frontend::AgentTicketPhone###DynamicField
- 値:

キー	内容
Field1	1
Field2	1
Field3	1
Field4	1

キー	内容
Field5	1
Field6	1
Field7	1

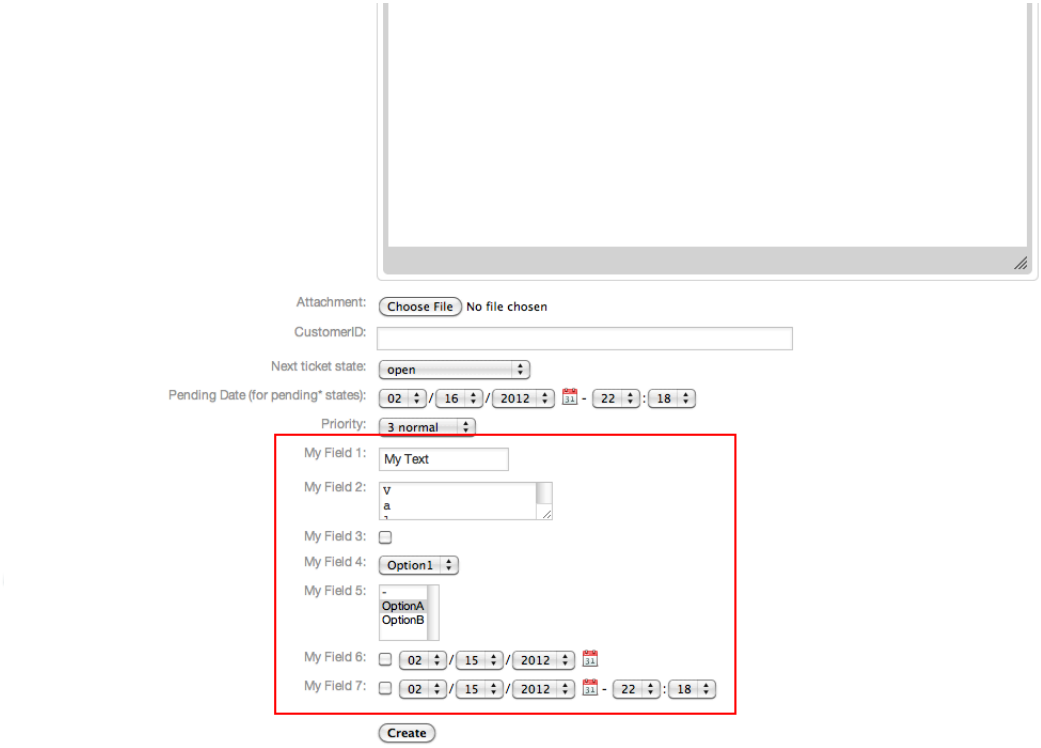
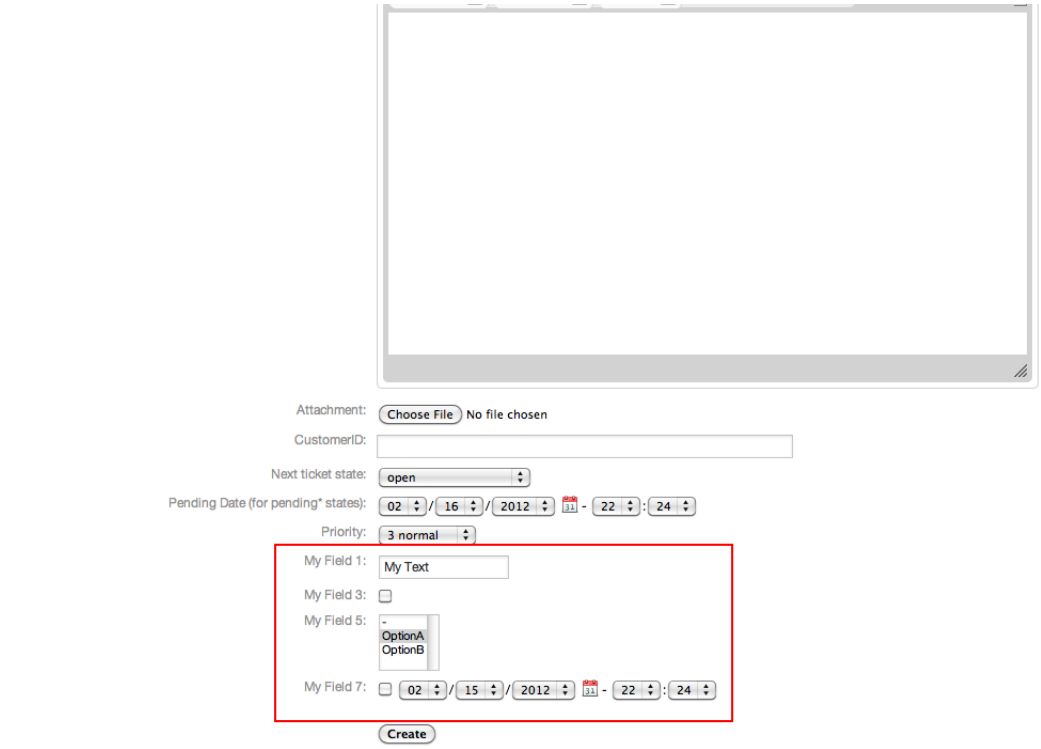


図:新規電話チケット画面中の必須の複数フィールド

例**4.24** 新規電話チケット画面中の複数の必須フィールドを非活性化

- グループ: チケット
- サブグループ: Frontend::Agent::Ticket::ViewPhoneNew
- 設定: Ticket::Frontend::AgentTicketPhone###DynamicField
- 値:

キー	内容
Field1	1
Field2	0
Field3	1
Field4	0
Field5	1
Field6	0
Field7	1



図：新規電話チケット画面中の必須の複数フィールドの非活性化

例**4.25** チケットズーム画面でフィールド 1 を活性化

- グループ: チケット
- サブグループ: Frontend::Agent::Ticket::ViewZoom
- 設定: Ticket::Frontend::AgentTicketZoom###DynamicField
- 値:

キー	内容
Field1	1

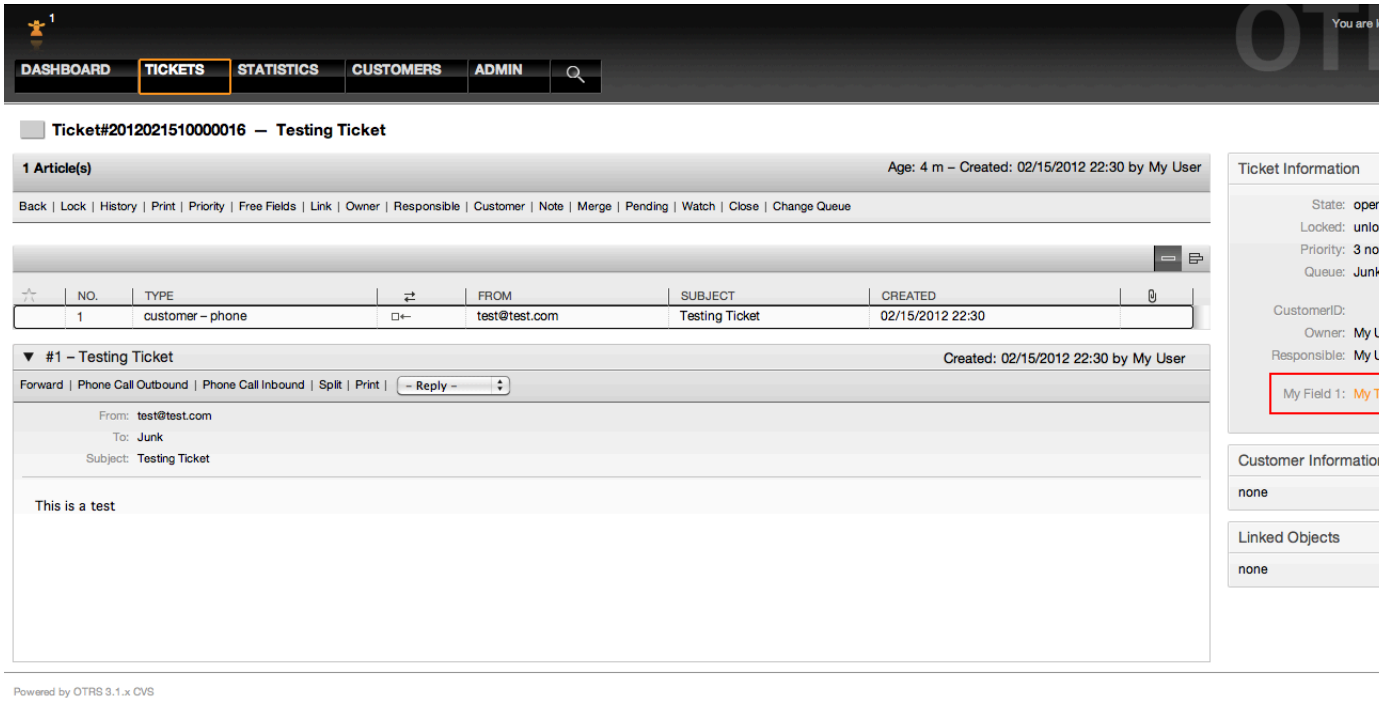


図:チケット・ズーム・画面中の*Field1*

例**4.26** チケット概要小画面の中の**Field1**を活性化

- グループ: チケット
- サブグループ: Frontend::Agent::TicketOverview
- 設定: Ticket::Frontend::OverviewSmall###DynamicField
- 値:

キー	内容
Field1	1

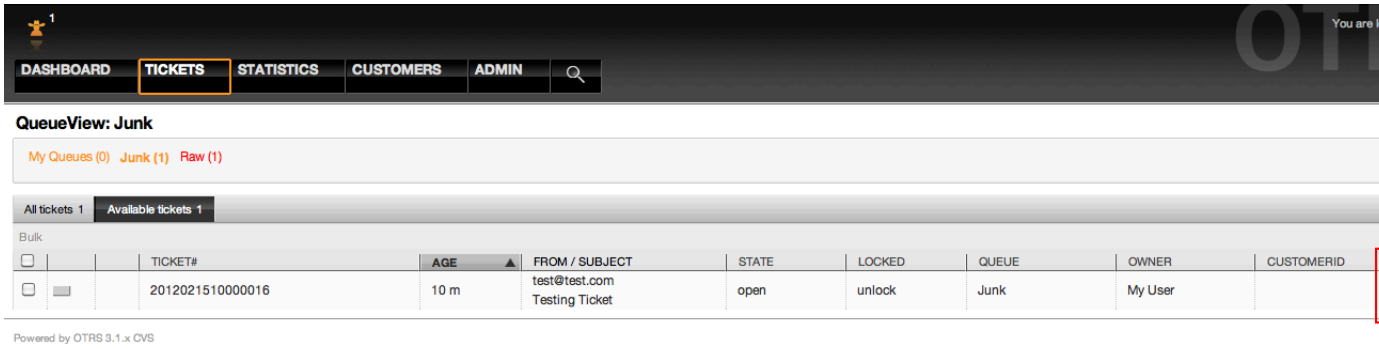


図:チケット概要小画面中のフィールド 1

このセッティングは、次のものに影響します:　　エスカレーションビュー、ロックされたビュー、キュービュー、信頼できるビュー、ステータスビューおよび監視ビュー画面。

10.2.11. チケット・イベント・モジュールによってデフォルト値を設定

フィールドが値をまだ持っていない場合、チケット・イベント(例えば、チケット作成)は、あるフィールド用の値セットを引き起こすことができます。

注記

この方法の使用によって、値がチケットの生成の後にセットされるので、このデフォルト値は編集画面(例えば新規電話チケット)で見られません。

この特徴を活性化するために、次のセッティングを可能にすることが必要です:"Ticket::EventModulePost###TicketDynamicFieldDefault".

例4.27 チケット作成イベントのフィールド 1 を活性化

- グループ: チケット
- サブグループ: Core::TicketDynamicFieldDefault
- 設定: Ticket::TicketDynamicFieldDefault###Element1

注記

この構成は、16のTicket::TicketDynamicFieldDefault###Element　セッティングのうちのどの中でもセットすることができます。

16フィールド以上をセットアップする必要があるなら、$OTRS_HOME/Kernel/Config/Files　ディレクトリにあるXMLファイルは置き換えてこの機能を拡張します。

- 値:

キー	内容
イベント	チケット作成
Name	Field1
値	新しい値

10.2.12. ユーザー選択によってデフォルト値をセット

動的フィールドデフォルト値は、ユーザー選択に格納されたユーザに定義された値で上書きすることができます。

この方法を使用して、フィールドのデフォルト値は、フィールドが活性化されている画面に表示されます。(フィールドが異なる値をまだ持っていない場合)

「Sysconfig」設定は、サブグループ「Frontend::Agent::Preferences」の「PreferencesGroups###DynamicField」に存在します。この設定は、選ばれたユーザのために排他的な動的フィールドデフォルト値をセットするために、ユーザー選択画面でエントリーを作る方法の例です。この設定の制限は、単にそれが1つの動的フィールドの使用を許すということです。2つ以上のフィールドがこの特徴を使用する場合、これに似ているより多くの設定を加える、カスタムXML設定ファイルを作成することが必要です。

注記

より多くのセッティングが新規XMLに追加され、各設定名がシステムの中でユニークかつ、"PreferencesGroups###DynamicField"と異なっていなければならないならば、覚えておいて下さい。例えば、PreferencesGroups###101-DynamicField-Field1, PreferencesGroups###102-DynamicField-Field2, PreferencesGroups###My-Field1, PreferencesGroups###My-Field2,等。

例4.28 ユーザー選択での**Field1**を活性化

- グループ: チケット
- サブグループ: Frontend::Agent::Preferences
- 設定: PreferencesGroups###101-DynamicField-Field1
- 値:

キー	内容
イベント	チケット作成
Active	1
Block	入力
Column	その他の設定
Data:	$Env{"UserDynamicField_Field1"}
Key:	My Field 1
Label:	My Field 1用のデフォルト値
Module:	Kernel::Output::HTML::PreferencesGeneric
PrefKey:	UserDynamicField_Field1
Prio:	7000

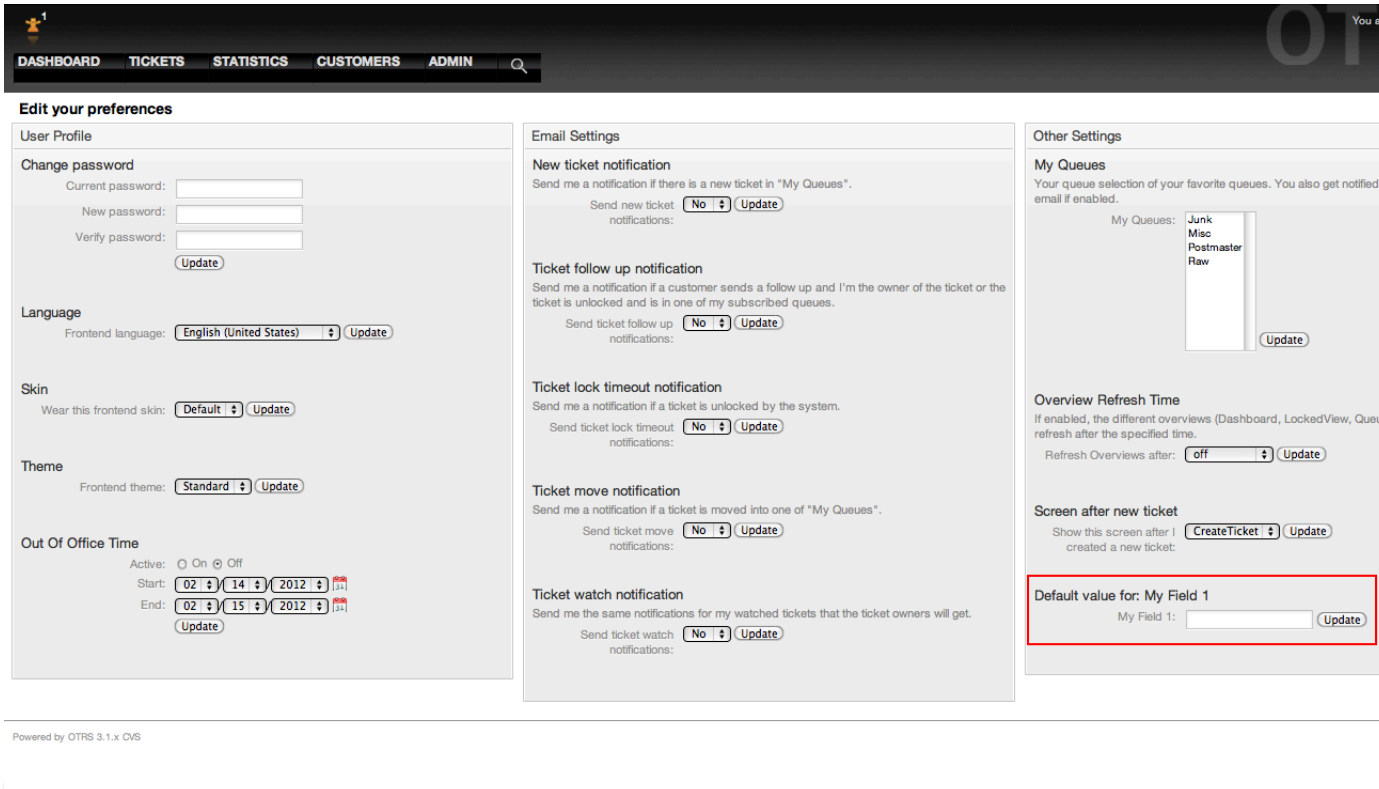


図:ユーザー選択画面中の*Field1*

11. 一般インターフェイス

OTRSの一般インターフェースは、OTRSをウェブサービスによって他のシステムと通信させる複合レイヤーのフレームワークから成ります。このコミュニケーションは双方向になりえます。:

- 供給者としての*OTRS:* OTRSは、外部システムからのリクエストを聞き、要求されたアクションを行なって、情報を処理し、リクエストに答えるサーバーとして働きます。
- 依頼者としての*OTRS:* OTRSは、遠隔のシステムにリクエストを送り、レスポンスを待って、クライアント情報収集者として働きます。

11.1. 一般インターフェース・レイヤー

一般インターフェースは柔軟で、かつカスタマイズするのが簡単になるようにレイヤー・モデルに基づいた構造である。

レイヤーは1セットのファイルである。それは、一般インターフェースがウェブサービスの異なる部分をどのように行なうかコントロールします。正しい構成を使用すると、一つは新規モジュールを作成せずに、異なる外部システムのために異なるウェブサービスを構成することができます。

注記

遠隔のシステムが一般インターフェースの現在の束ねられたモジュールを支援しない場合、その特定のウェブサービスのために特別のモジュールを開発している必要がある。

OTRS出荷時に提供される一般インタフェース・モジュールのリストは今後更新され増加していくでしょう。

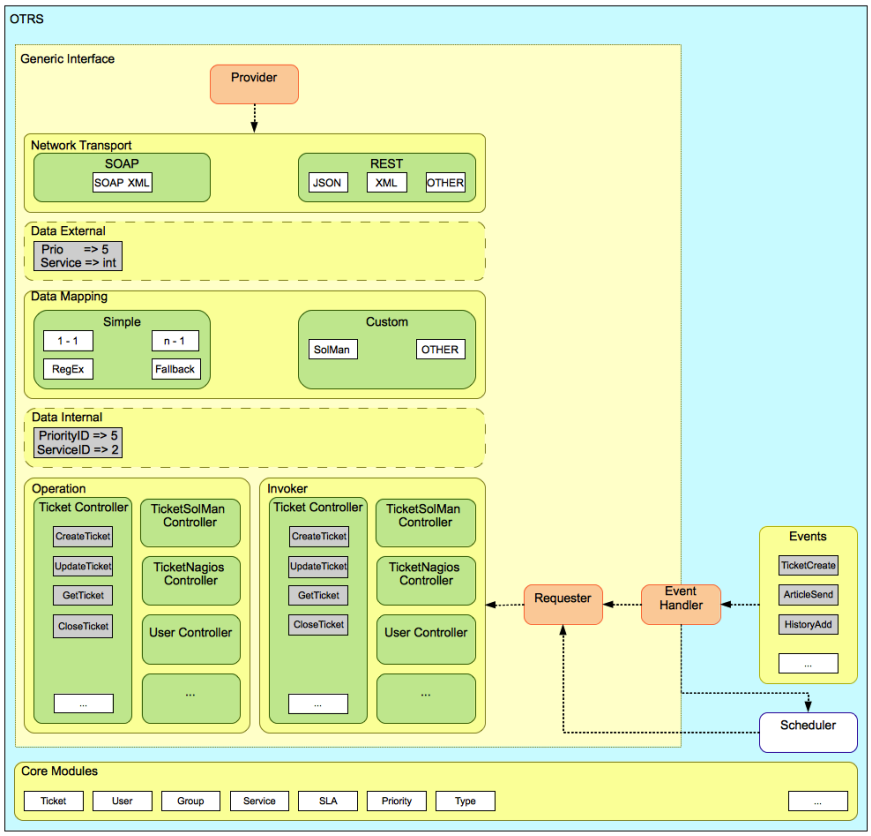


図:グラフィカル・インターフェース・レイヤー

11.1.1. ネットワーク・トランスポート

このレイヤーは遠隔のシステムとの正確なコミュニケーションに責任を持ちます。依頼を受け取り、供給者として働く場合、レスポンスを生成し、依頼者として働く場合、依頼および受信応答を生成します。

供給者コミュニケーションは「nphgenericinterface.pl」と呼ばれる新規ウェブサーバによって取り扱われます。

依頼者コミュニケーションは、一般インタフェース・モジュールあるいは他のOTRSモジュールが引き金となって起きたイベント中に、コミュニケーションを開始することができるかもしれません。このイベントははイベント・ハンドラーによって捕捉されます。構成上、イベントは依頼者オブジェクトまたはスケジューラ専用によって直接処理されます(別々のデーモンが非同期的にタスクを処理しようとします)。

11.1.2. データマッピング

このレイヤーは、データ構造とOTRSそしてリモートシステムの間の翻訳について責任を持ちます(データ内部およびデータ外部層)。通常遠隔のシステムはOTRS(異なる値およびそれらの値の名前を含む)とは異なっているデータ構造を持っています。そしてここに受信情報を「OTRSが理解する何かへと変更するレイヤーの重要性が存在します。また、逆にデータ辞書を使うリモートシステムへの送信情報についてもです。

例:「優先度」(OTRS)は遠隔のシステムでの「Prio」と呼ばれるかもしれません。また、それは、値「1 低」(OTRS)が遠隔のシステムについての「情報」にマッピングされるべきです。

11.1.3. コントローラー

コントローラーは同様のオペレーションあるいは依頼アクション(Invokers)の集合です。例えば、チケット・コントローラーはいくつかの標準チケット・オペレーションを含んでいます。カスタム・コントローラは実装可能です。例えば「チケット外部会社」コントローラーは標準チケット・コントローラーと同様の機能を含みます。しかし異なるデータ・インターフェイスまたは機能名（リモート・システム機能名に適応する）または異なるコードを完成します。

一般インターフェースのための1つのアプリケーションは、同じ種類のもう一つのリモートシステムと話すことだけができるようなリモートシステムと情報を同期できるようになります。この場合、新規コントローラーは開発される必要があります。そして、オペレーションと依頼アクション(Invokers)は、リモートシステムの振る舞いをエミュレートしなければいけません。その振る舞いは、OTRSが提供するインターフェイスがリモートシステムのインターフェイスと同様です。

11.1.4. オペレーション(供給者としてのOTRS)

オペレーションはOTRSの内に行なうことができる単一の行為です。オペレーションにはすべて同じプログラミングインターフェースがあります、それらは1つの特定のパラメタへデータを受け取り、成功ステータス、潜在的なエラーメッセージおよび返るデータを持ったデータ構造を返します。

通常、オペレーションは、コアモジュールを呼び出すために、すでにマップされたデータ（内部）を使います。そしてOTRSでアクション（チケット作成、ユーザー更新、キューの無効か、通知送信）を実行します。オペレーションはアクションを実行するOTRS　APIへのフルの権限を持っています。

11.1.5. 依頼アクション (依頼者としてのOTRS)

依頼アクション（Invokers）とは、OTRSがリモートシステムに対して行うアクションのことです。依頼アクション(Invokers)は、リクエストを作成する必要とされる情報を処理し収集するためにOTRSコア・モジュールを使用します。情報が準備ができている場合、リモート・システムに送るためにリモート・システムのフォーマットにそれをマッピングしなければなりません。それはアクションを実行する情報を処理し、成功を処理するためか、エラーを取り扱うためのいずれかのために返答を送信します。

11.2. ウェブサービス・コマンドライン・インタフェース

一般インターフェイスは、供給者としてそして同様に要求者としてアクションを実施するために定義済フローをも待っています。

これらのフローは以下に記述されます。

11.2.1. 供給者としてのOTRS

11.2.1.1. リモートリクエスト：

1. HTTPリクエスト
   - OTRSは、HTTPリクエストを受け取りそれをレイヤーに渡します。
   - 供給者モジュールはこれらのアクションを実行しコントロールすることを担当しています。
2. ネットワーク・トランスポート
   - ネットワーク・トランスポート・モジュールは、データ・ペイロードをデコードし、オペレーション名とデータのそれ以外の部分とを区別します。

- オペレーション名およびオペレーションデータは供給者に返されます。

3. データ外部
   - リモートシステムから送信されたデータ（これはモジュールベースのレイヤーではありません）。
4. マッピング
   - データは、このオペレーション(入信する依頼データ用マッピング)のためのマッピング構成で指定されるように、外部システム・フォーマットからOTRS内部形式へ変換されます。
   - 既に変換されたデータは、供給者に返されます。
5. データ内部
   - 変換され、そしてオペレーション(これはモジュールベースのレイヤではありませんん)に渡される準備が整ったデータ。
6. オペレーション
   - データを受取り、有効化します。
   - ユーザ・アクセス・コントロールを行います。
   - アクションを実行します。

11.2.1.2. OTRSレスポンス:

1. オペレーション
   - 供給者に結果データを返します。
2. データ内部
   - オペレーションから返されるデータ。
3. マッピング
   - データは、マッピング構成(発信の応答データ用マッピング)で指定されるリモート・システム・フォーマットに変換されます。
   - 既に変換されたデータは、供給者に返されます。
4. データ外部
   - 返答としてネットワーク・トランスポートに渡されるために変換され、準備されるようなデータ。
5. ネットワーク・トランスポート
   - 既にリモート・システム・フォーマット済みのデータを受け取ります。
   - このネットワーク・トランスポートタイプのための有効なレスポンスを構成します。
6. HTTPレスポンス
   - 応答はウェブサービス・クライアントに返送されます。
   - エラーの場合には、エラー・レスポンスが、遠隔のシステム(例えばSOAPエラー、HTTPエラーなど)に送られます。

11.2.2. 依頼者としてのOTRS

11.2.2.1. OTRS依頼:

1. イベント・トリガー・ハンドラ
   - ウェブサービス構成に基づいて、依頼が同期か非同期かを決定します。
     - 同期
       - 依頼者への直通電話は、新規依頼を作成し、それをレイヤーを通して渡すためにされます。
     - 非同期
       - OTRSスケジューラのための新規一般インターフェース(依頼者)タスクを作成します。（スケジューラーに依頼の実行を委任することによって、ユーザー体験は非常に改善されるかもしれません。そうでなければ、常に依頼を準備する必要になってしまいます。またリモート実行はそれらの依頼のきっかけとなるOTRSエベントに追加されます。）
       - その次のサイクルでは、スケジューラ・プロセスは新規タスクを読んで、新規依頼を作成し、それをレイヤーへ渡す依頼者への呼び出しを作成します。
2. 依頼アクション(Invokers)
   - イベントからデータを受け取ります。
   - 受信データ(もし必要ならば)を有効にします。
   - データを補足する(もし必要ならば)呼び出しコア・モジュール。
   - 依頼データ構造を返すか、あるいは卒なく依頼を取り消すために依頼者にStop Communication信号を送ってください。
3. データ内部
   - 依頼アクション(Invokers)(これはモジュールベースのレイヤではありません)から渡されるようなデータ。
4. マッピング
   - データは、マッピング構成(発信の応答データ用マッピング)で指定されるリモート・システム・フォーマットに変換されます。
   - 既に変換されたデータは、依頼者に返されます。
5. データ外部
   - 変換され、遠隔のシステムに送るために準備できたデータ。
6. ネットワーク・トランスポート
   - 依頼者からリモート・システム表示形式に既に変換されたリモート・オペレーション・フォーマットおよびデータを受け取ります。
   - ネットワーク・トランスポートの有効な依頼を構成します。
   - 遠隔のシステムにリクエストを送り、応答を待ちます。

11.2.2.2. リモート応答:

1. ネットワーク・トランスポート

- レスポンスを受け取り、データ・ペイロードをでコードします。
- 依頼者にデータを返します。

2. データ外部
   - リモート・システムから受け取るデータ。
3. マッピング
   - データは、このオペレーション(入信応答データ用マッピング)のためのマッピング構成の中で指定される外部システム・フォーマットからOTRS内部形式へと変換されます。
   - 既に変換されたデータは、依頼者に返されます。
4. データ内部
   - 既に変形済データは、依頼者に返されます。
5. 依頼アクション(Invokers)
   - 返信データを受け取る
   - 特に各依頼アクション(Invokers)(もしあればエラーハンドリングを含む)に必要とされていたデータを処理します。
   - 依頼者に依頼アクション(Invokers)結果およびデータを返します。
6. イベント・ハンドラあるいはスケジューラ
   - 依頼者からのデータを受け取ります。スケジューラの場合、このデータは、タスクを直ぐにまたは将来に再スケジュールするような情報を含んでいるかもしれません。

11.3. ウェブサービス

Webサービスは、私たちの場合、OTRSとリモート・システムとの間のコミュニケーション方法です。

Webサービスの核心はその構成(Configuraiton)です。尚、ここで、以下が定義されます。ウェブサービスがどのアクションを内部に行なうことができるか(オペレーション)、どんなアクションが、OTRS要求がリモート・システムを実行できるか（依頼アクション(Invokers)）どのようにデータが他からひとつのシステムへ変換されるか（Mapping）そしてどのプロトコル上でコミュニケーションが発生するのはどのプロトコル上か（Transport）。

一般インターフェースは、互いから独立して、交換可能な、既に作られていたビルディングブロックを使用して、事前定義された方法でOTRSのためのWebサービスを作成することを可能にするフレームワークです。

11.4. ウェブサービス・グラフィカル・インターフェース

ウェブサービスGUI(GUI)は、ユーザー・フレンドリーで便利なインターフェースで複雑なウェブサービス構成を構築することを可能にするツールです。それは次のものに許可します:

- ウェブサービスを作成および削除します。
- 既存のウェブサービスのためのエクスポート/インポート構成(YAMLファイルフォーマット中の)。
- ビュー、ウェブサービスで既存のウェブサービス用の復旧と旧構成のエクスポート

- デバッガ画面中の各ウェブサービスのすべてのコミュニケーション・ログを追跡してください。

11.4.1. ウェブサービス概要

管理者インターフェース(システム管理ボックス中の)のメイン画面中の「ウェブサービス」リンクは、ウェブサービス概要画面に向かいます。そこでは、ウェブサービス構成を管理することができます。新規ウェブサービスを加えるか、あるいはこの画面からの既存の構成を変更することができます。

ウェブサービス構成画面は、その上部に案内のために「パンくずリスト」を持っています。この案内はウェブサービス構成のどの部分に自分がいるのかを正確に知るために便利です。
　また、その案内は、さらにユーザが構成プロセスの任意の部分にいつでも(このアクションは変更を保存しないでしょう)ジャンプすることを可能にします。

注記

新規ウェブサービスを作るために、「ウェブサービスの追加」ボタンを押し、必須情報を提供します。

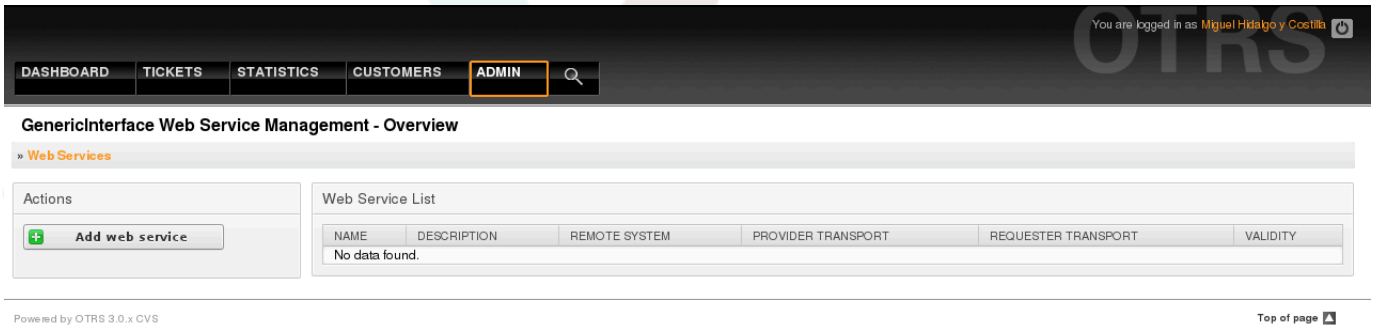


図:ウェブサービス概要

11.4.2. ウェブサービスの追加

この部分中の必要な唯一のフィールドは、システムにおいて独特である必要があり、空にしておくことができないウェブサービス「名」です。さらに、他のフィールドは「デバッグのしきい値」および「有効性」のように構成に必要であるが、これらのフィールドはすでに各リスト用にデフォルト値でたくさんあります。

「デバッグのしきい値」のためのデフォルト値は「デバッグ」します。このように構成された時、全てのコミュニケーションログがデータベースで登録されます。続いて起こるデバック閾値はより制限的で　システム内の１セットより低く古いコミュニケーションログを破棄します。

デバッグ閾値レベル (下から上へ)

- Debug
- Info
- 通知
- Error

さらに、「供給者としてのOTRS」および「依頼者としてのOTRS」のためのネットワーク・トランスポート・プロトコルを定義することは可能です。

データベースの中で新規ウェブサービスを登録するために「保存」ボタンをクリックするか、またはこのオペレーションを廃棄するために「キャンセル」をクリックして下さい。ウェブサービス概要画面に戻ります。

YAMLフォーマットの中に既にウェブサービス構成ファイルを持っていれば、画面の左側の「インポート・ウェブサービス」ボタンをクリックすることができます。ウェブサービスのインポートについてより詳細には、次のセクション「ウェブサービス変更」をチェックしてください。

注記

ウェブサービスに、より多くの詳細を変更・追加するためには、ウェブサービス概要画面中のウェブサービス名をクリックしてください。

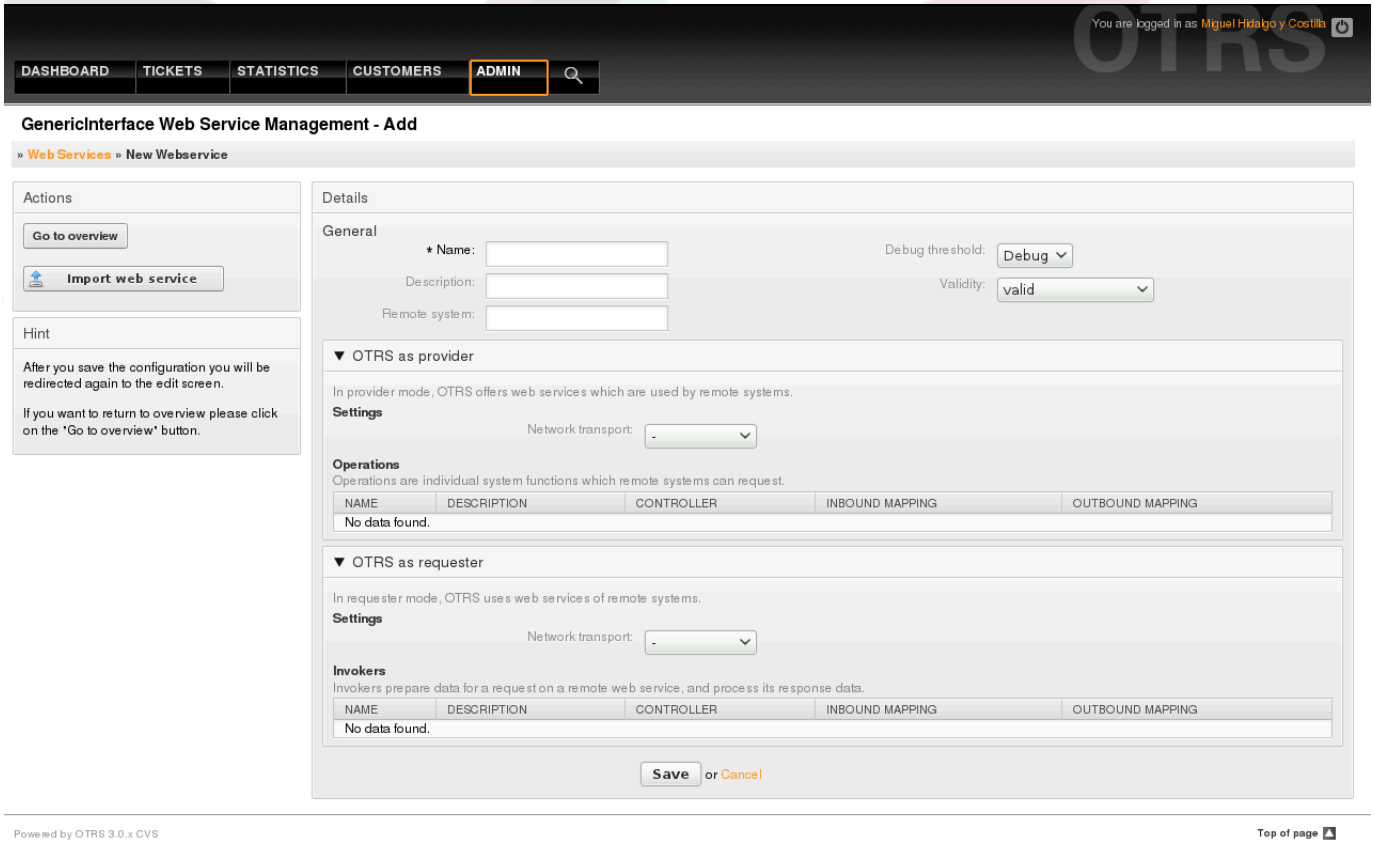


図:ウェブサービスの追加

11.4.3. ウェブサービス変更

この画面においては、ウェブサービスのすべての部分を扱う機能の完全セットを持っています。アクション・カラムの左側に

いくつかのボタンがあります。それによりウェブサービス上で全ての可能なアクションを実行できます。

- Clone web service.(クローン・ウェブサービス)
- Export web service.(エクスポート・ウェブサービス)
- Import web service.(インポート・ウェブサービス)
- Configuration History.(構成履歴)
- Delete web service.(ウェブサービスの削除)
- Debugger.(デバッガ)

注記

「構成履歴」および「デバッガ」はあなたを異なる画面へ導きます。

11.4.3.1. ウェブサービスのクローン

ウェブサービスのクローンを作るために、「クローンのウェブサービス」ボタンをクリックする必要があります。ダイアログは、あなたが使用できるデフォルト名またはウェブサービス用の新規名を表示します。

注記

ウェブサービスの名前が間違いなくシステム内にユニークであることを忘れないでください。

ウェブサービスのクローンを作成するために「クローン」ボタンをクリックして下さい。またはダイアログを閉じるために「キャンセル」をクリックして下さい。

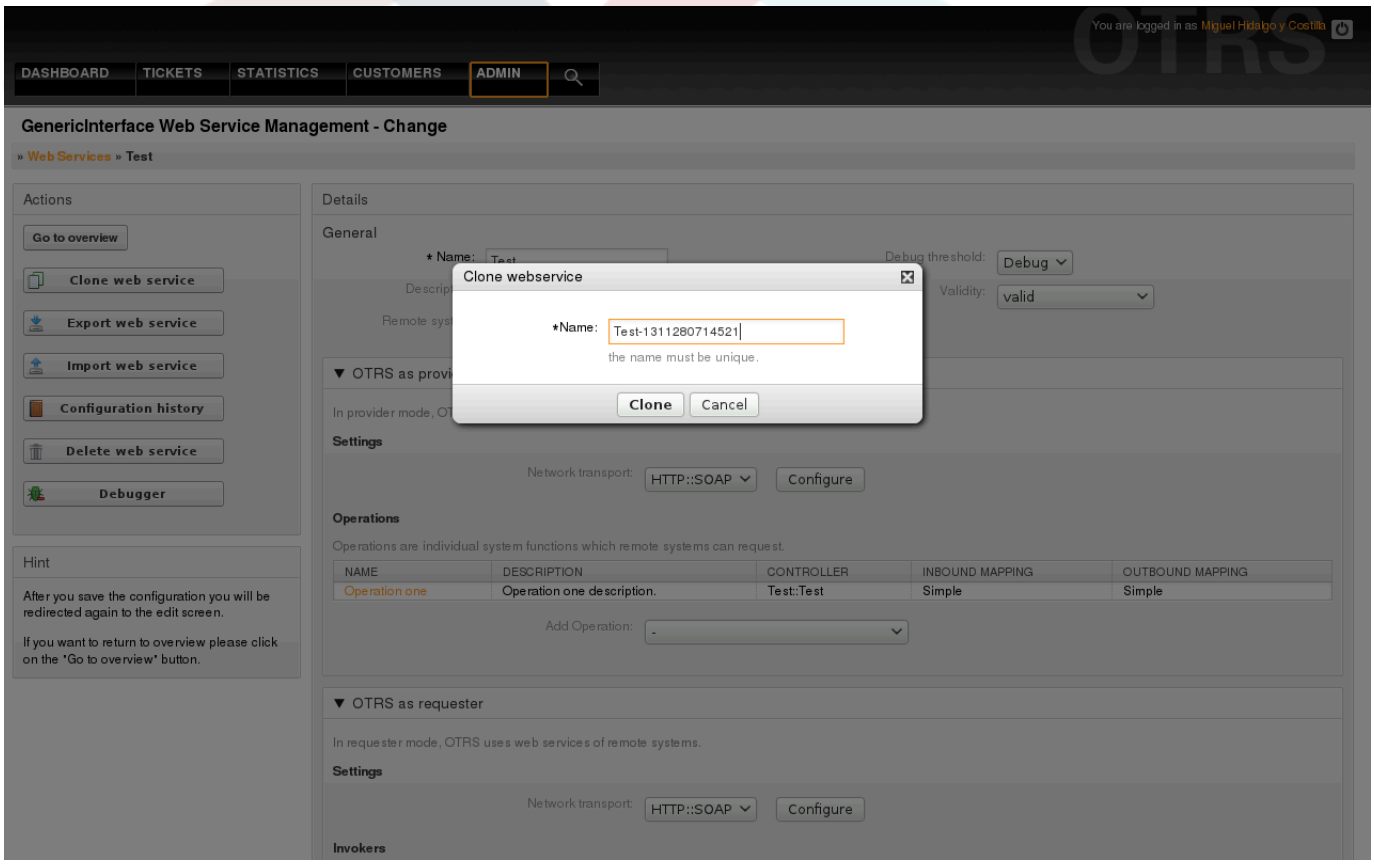


図:ウェブサービスのエクスポート

11.4.3.2. ウェブサービスのエクスポート

「エクスポート　ウェブサービス」ボタンは、YAMLファイルの中への現在のウェブサービスの構成をダンプし、それをダウンロードし、ファイルシステム上にそれを格納する機会を与えます。ウェブサービスを統合したければこれは特に有用です。（ひとつのサーバからもうひとつのサーバへ。例えばテスト環境から生産システムへ）

警告

ウェブサービス構成中の格納されたパスワードはすべて、プレインテキスト・フォーマットでエクスポートされます。

「エクスポート・ウェブサービス」をクリックする直後、ちょうどウェブページ上のファイルダウンロードリンクでクリックする時のようにブラウザの保存ダイアログが現れます。

注記

各オペレーティング・システム上のブラウザにはそれぞれそれ自身の保存ダイアログ画面およびスタイルがあります。ブラウザとその構成によってはダイアログが示されないことはありえます。また、ファイルはデフォルト・ディレクトリにあなたのファイルシステム上に保存されます。必要な場合はより特殊な指示があるかブラウザ・ドキュメンテーションをチェックしてください。

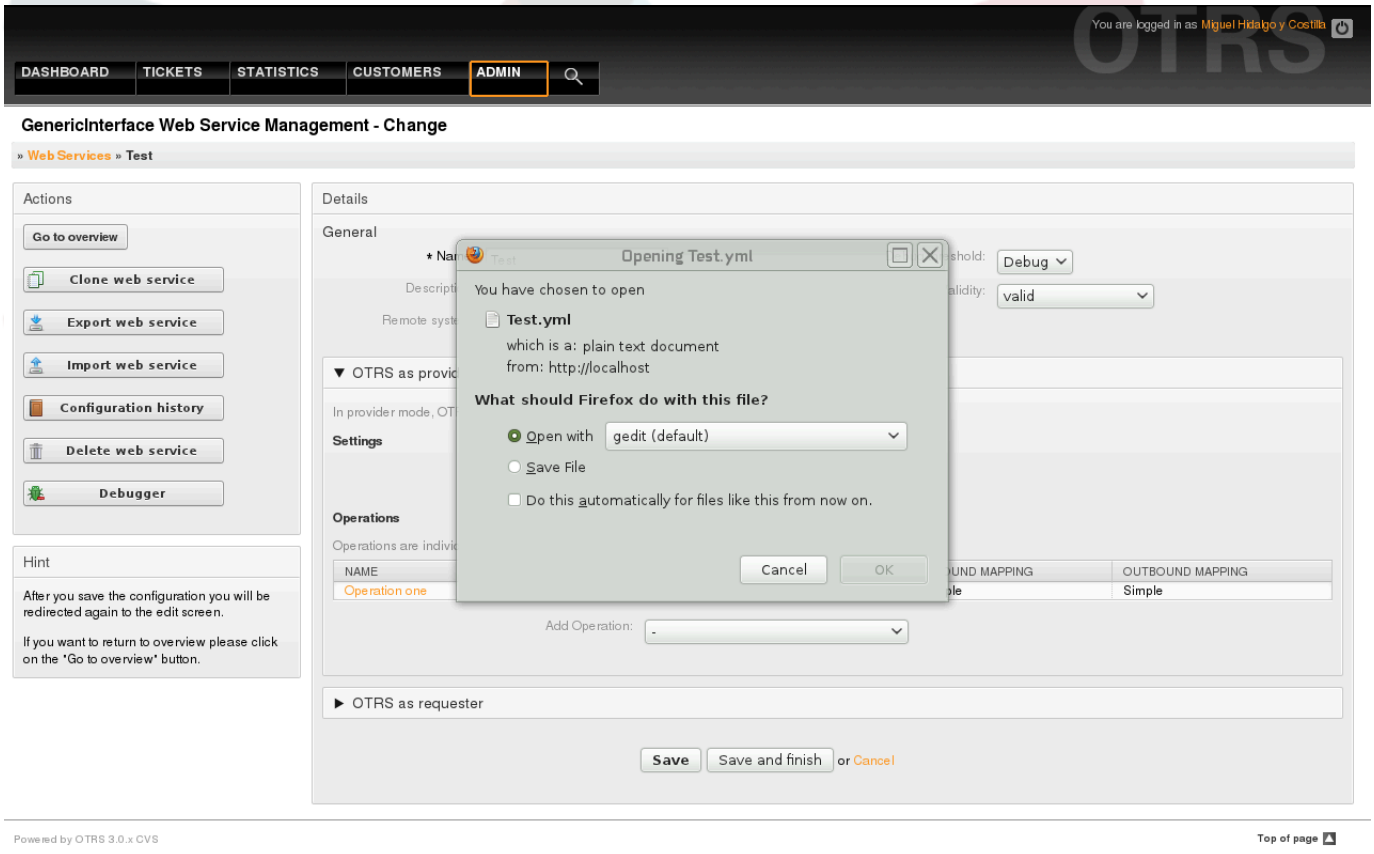


図:ウェブサービスのエクスポート

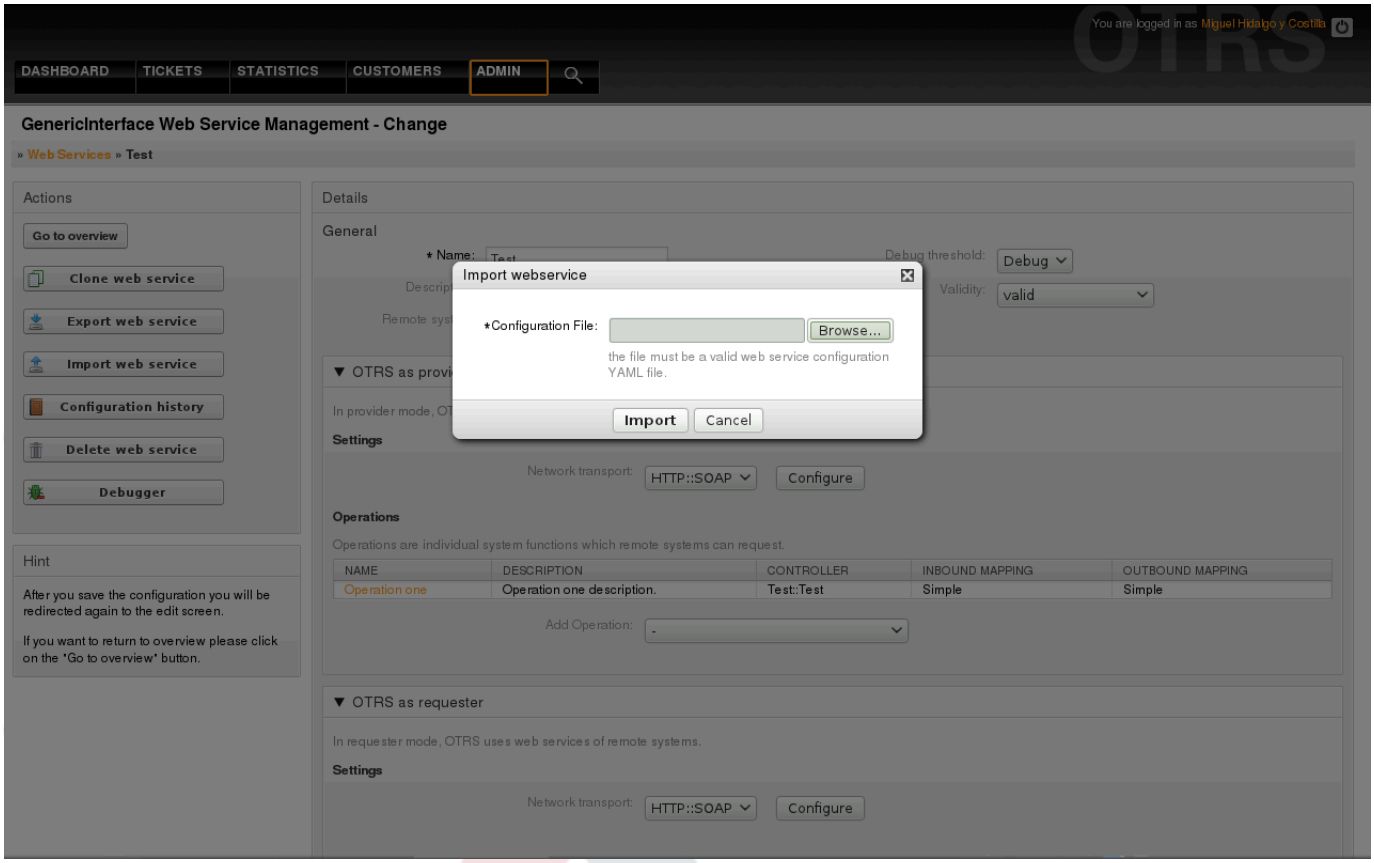
11.4.3.3. ウェブサービスのインポート

有効なウェブサービス構成YAMLファイルは、インポート・ウェブサービス機能を使用するように要求されます。「インポート・ウェブサービス」ボタンをクリックし、構成ファイルを探してブラウズし、入力ボックス中の完全なパスを提供してください。

ファイルから新規ウェブサービスを作るために「インポート」ボタンをクリックするか、あるいはダイアログを閉じるために「キャンセルしてください。」

注記

ウェブサービス名は、構成ファイル名から得られます。(例えば、ファイル名がMyWebservice.ymlならば、結果として生じるウェブサービスはMyWebserviceと命名されます。)ウェブサービスが、あなたがインポートしたいウェブサービスと同じ名前でシステムで登録されれば、システムは、あなたにインポートウェブサービスの名前を変更させるためにウェブサービス変更画面に導くでしょう。



図：ウェブサービスのインポート

11.4.3.4. ウェブサービス履歴

ウェブサービス構成へのすべての変更はウェブサービス履歴(ジャーナルとしての)に新規エントリーを作ります。ウェブサービス履歴画面は、ウェブサービスのためのすべての構成バージョンのリストを表示します。「構成履歴の一覧」の中の列(バージョン)はそれぞれ、ウェブサービス履歴に一度の修正を表わします。

それが特定の日付/時刻であれば、全体の構成を示すには行のうちの1つをクリックしてください。構成はこの画面の「履歴詳細」セクションの中で示されるでしょう。ここでは、選択されたウェブサービス構成バージョンをエクスポートすることも、あるいは現在のウェブサービス構成へそのバージョンを回復することもできます。

「エクスポート・ウェブサービス構成」は、ウェブサービス変更画面中の「エクスポート・ウェブサービス」機能として正確に作用します。より詳細には、そのセクションを参照します。

現在のウェブサービス構成への変更が予想通りに働かず、変更を手動で復帰するのが簡単でない場合、「ウェブサービス構成を復帰」ボタンをクリックすることができます。これは、あなたが必ずウェブサービス構成を復帰させるかどうかあなたに尋ねるためにダイアログを開始するでしょう。

現在の選択したバージョンで構成を代替するためにこのダイアログで「ウェブサービス構成を復帰」をクリックするか、あるいは、「キャンセル」をクリックしてダイアログを閉じます。

警告

ウェブサービス構成に格納されたどんなパスワードもプレインテキスト・フォーマットでエクスポートされるだろうということを覚えておいてください。

構成を回復する場合は注意してください。このプロセスは不可逆です。

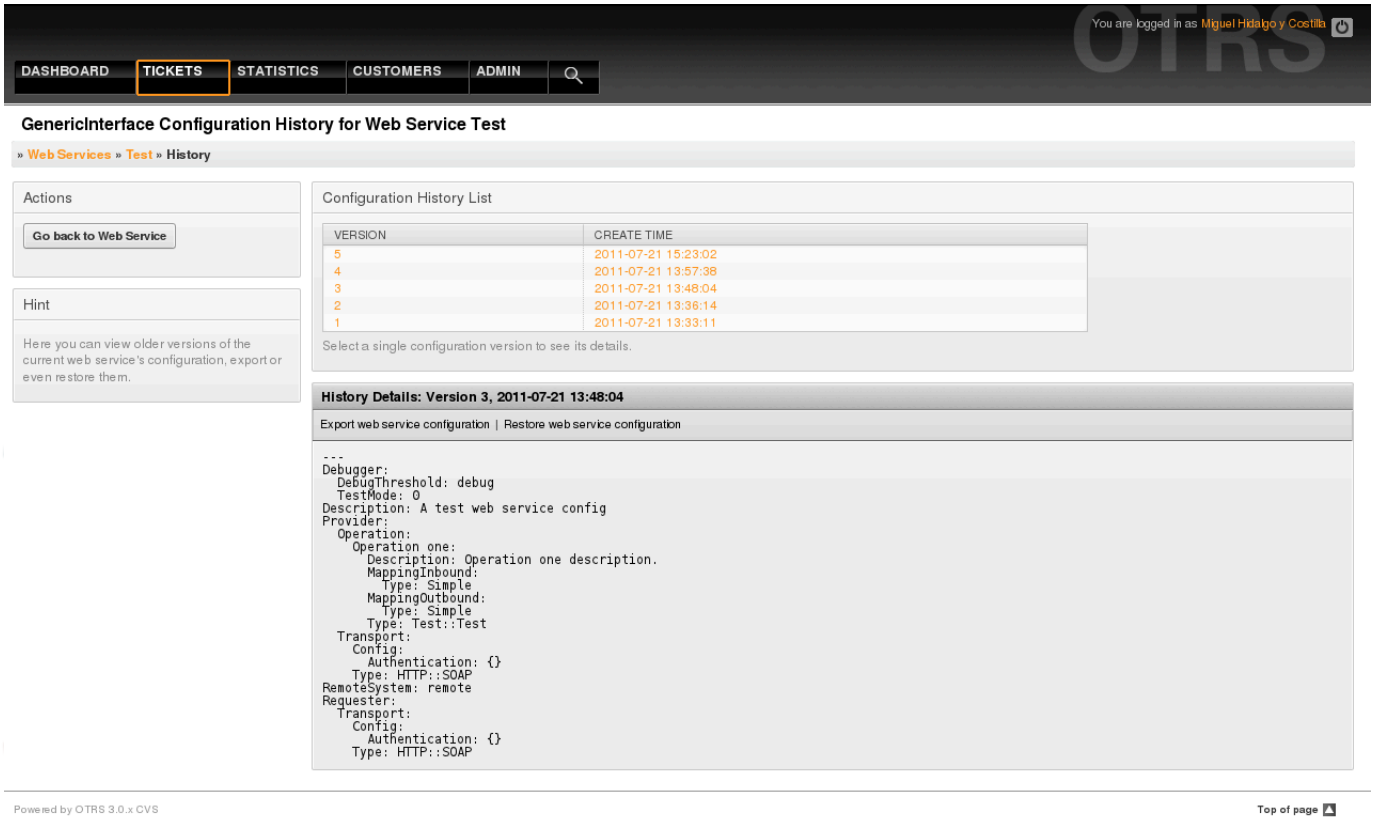


図:ウェブサービス履歴

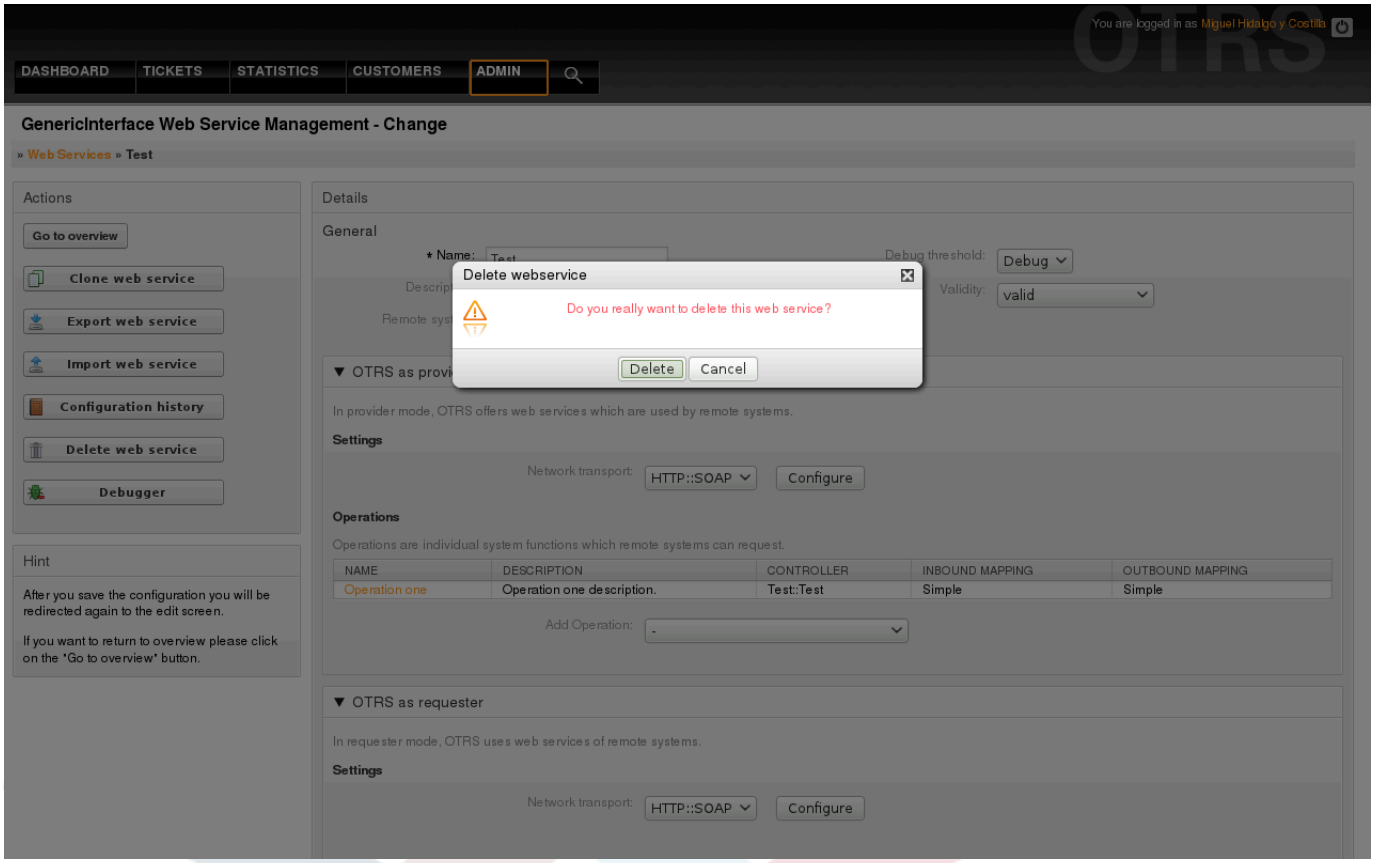
11.4.3.5. ウェブサービスの削除

時としてウェブサービスを完全に削除することが必要な場合があります。これをするために、あなたは「ウェブサービスの削除」ボタンを押すことができます。すると、新規ダイアログが確認を求めて現われるでしょう。

ウェブサービスの削除を確認するために「削除」をクリックするか、またはダイアログを閉じるために「キャンセル」をクリックして下さい。

警告

ウェブサービスの削除はやり直しがききません。ウェブサービスを削除する場合、注意してください。



図：ウェブサービスの削除

11.4.3.6. ウェブサービス・デバッガ

デバッガは、ウェブサービスのログを格納します。デバッガ画面では、供給者あるいは依頼者タイプのいずれかのためのウェブサービス・コミュニケーションをすべて追跡することができます。

この画面が示される場合、依頼リストがロードし始めます。リストが完全に満たされた後、その詳細をチェックするために列のうちの1つを選ぶことができます。(列とはコミュニケーション・シーケンスを意味します)これは詳述します、下のボックスに現われます。

画面の右側ではフィルタを使用して、コミュニケーション・リストを狭くすることができます。次のものによってフィルタすることができます：

- コミュニケーション・タイプ(供給者または依頼者)
- 日付:前、及び/または特定の日付の後
- リモートのIPアドレス
- 上記すべての組み合わせ

フィルタ設定がセットされた後、「リフレッシュ」ボタンを押してください。検索基準を満たす新規リストが表示されるはずです。

注記

フィルターの検索基準に依存しますが、新規リストが結果を返さないこともあり得ます。

アクション・カラムの下の画面の左の側では、「ウェブサービスに戻る」を選択するか、あるいは「クリア」ボタンを押すことによりデバッガログを取り除くことができます。ダイアログが開い

て、ログの削除を確認されます。アクションを実行するためにダイアログボタンで「クリア」をクリックするか「キャンセル」をクリックしてこのダイアログを閉じます。

「リクエスト詳細」セクションで、選択されたコミュニケーションの詳細をすべて見ることができます。ここでは、完全なフローを追跡し、起こりうるエラーをチェックし、あるいは成功の応答を確認することができます。

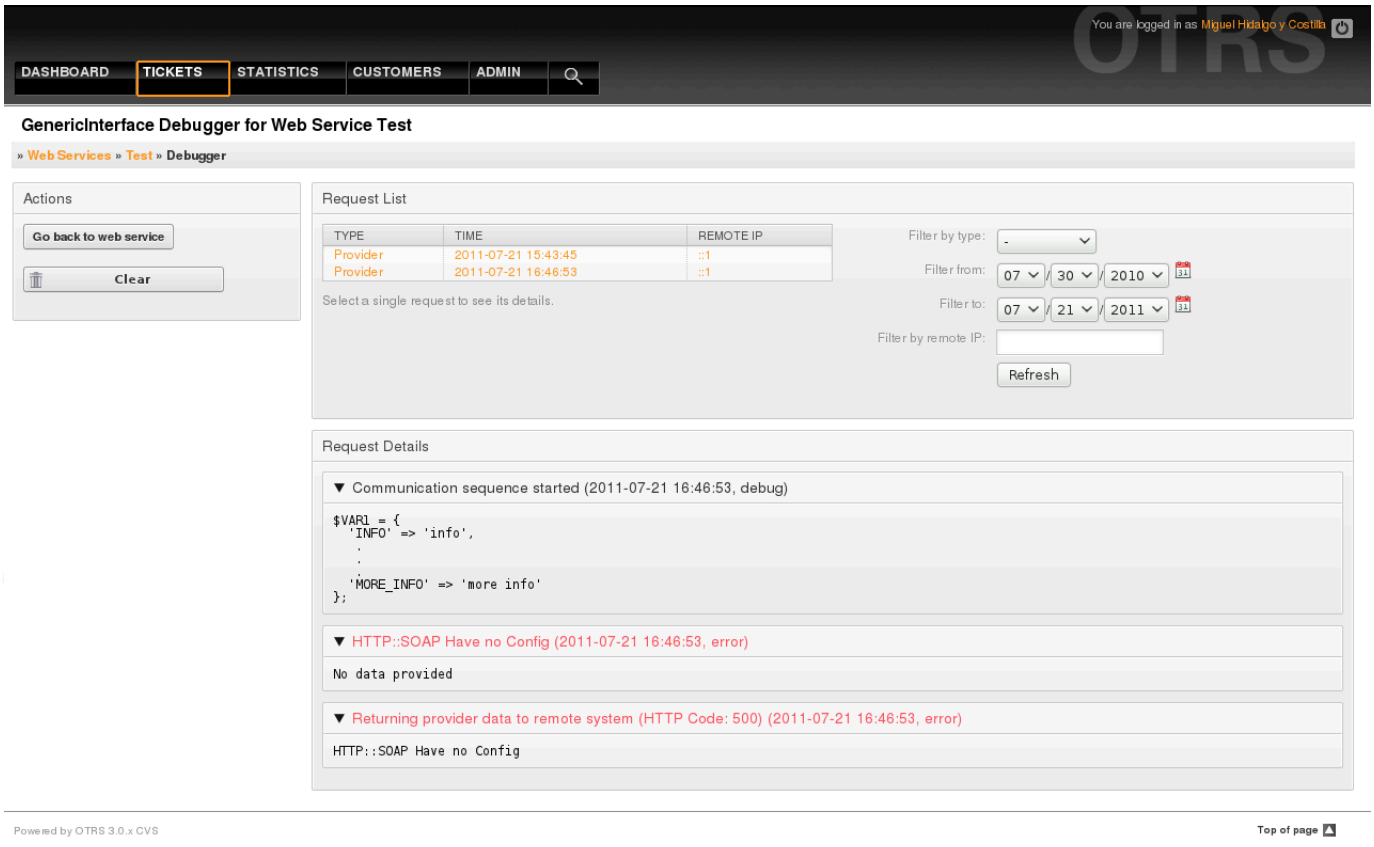


図:ウェブサービス・デバッガ

11.4.3.7. ウェブサービス構成変更

ウェブサービス変更画面に戻って、その右側をレビューしましょう。ここでは、ウェブサービスのためのすべての一般データを修正することができます。一般データとはたとえば名前、記述、デバッグ閾値などです。さらに下には二つ以上の選択があります。それによって、「供給者としてのOTRS」や「依頼者としてのOTRS」といったコミュニケーション・タイプ用の特定のパラメタを変更することができます。

ウェブサービス構成は各レベル上に保存する必要があります。このことは設定が変われば、現在の構成レベルを保存してしまうことで、他へのリンクや構成の深い部分が無効にしてしまうことを意味します。保存後に、無効となったリンクは
再び構成を継続することを可能にすることで、再度可能にできます。

「供給者としてのOTRS」セクションにおいては、ネットワーク・トランスポート・プロトコルをセットするか構成することが可能です。登録されるネットワーク・トランスポート・バックエンドだけがリストに示されます。ネットワーク・トランスポートを構成するためには「構成」ボタンをクリックします。さらに、このボックスの中に新規オペレーションを追加することは可能です。これを行うするためには「オペレーションを追加」リストから可能なオペレーションをひとつ選択します。これはあなたをオペレーション構成画面へ導くでしょう。新規オペレーションを保存した後に、それは上記のテーブルにリストされるでしょう。

「依頼者としてのOTRS」は前のものに非常に似ています。しかし、「オペレーション」の代わりに、ここで依頼アクション(Invokers)を加えることができます。

ウェブサービスの構成を保存し継続するために「保存」ボタンをクリックするか、あるいはウェブサービス概要画面に保存し返るために「保存して終了」をクリックするか、現在の構成レベル変更を廃棄し、かつウェブサービス概要画面に返るために「キャンセル」をクリックして下さい。

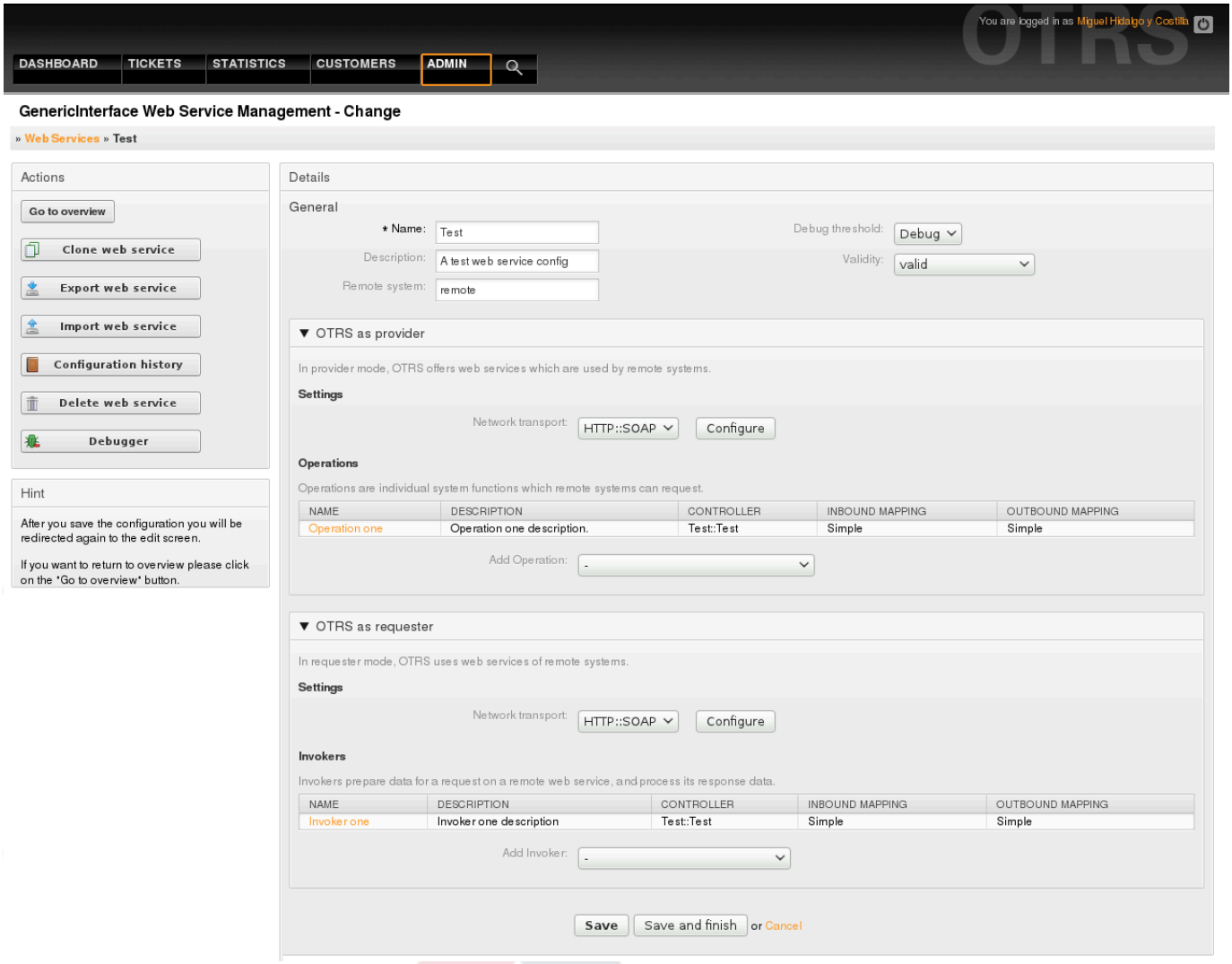


図:ウェブサービスの変更

注記

他の一般インターフェイスの構成画面のように（ネットワーク・トランスポート、オペレーション、依頼アクション(Invokers)およびマッピング）初期の構成（追加）画面は単にふたつのオプションを表します。「保存」と「キャンセル」です。構成画面に再度戻ると、新規オプション「保存および終了」が現われます。この特徴の振る舞いは下に定義されます。

「保存」はデータベースに現在の構成レベルを格納するでしょう。そして、それは、あなたが行った変更をレビューする、あるいはより深い設定を構成するために、前画面に戻ります。

「保存および終了」はデータベースに現在の構成レベルを格納します。また、それは、構成階層中の前画面(直上部の構成レベルへ)に戻ります。

「キャンセル」は、現在の構成レベルへのどんな構成変更も廃棄し、構成階層中の前画面に戻ります。

11.4.3.7.1. ウェブサービス供給者のネットワーク・トランスポート

今後、利用可能なネットワーク・トランスポートのリストは増加するでしょう。現在、「HTTP::SOAP」トランスポートだけが利用可能です。各トランスポートはセットアップのために異なる設定オプションを持っています。また、それらは、それを構成するために異なるフロント・エンド・モジュールを使用するかもしれません。しかし、ほとんど、それらは「HTTP::SOAP」トランスポート構成モジュールに類似しているべきです。

供給者として「HTTP::SOAP」プロトコルを構成することは全く簡単です。設定は二つのみです。:「ネームスペース」および「最大のメッセージ長」。これらのフィールドは必須です。最初のものは、曖昧さを少なくする文脈に、SOAP方法を与えるURIです。また、第2のものは、OTRSが処理する　SOAPメッセージのために最大サイズ(バイトで)を指定することができるフィールドです。

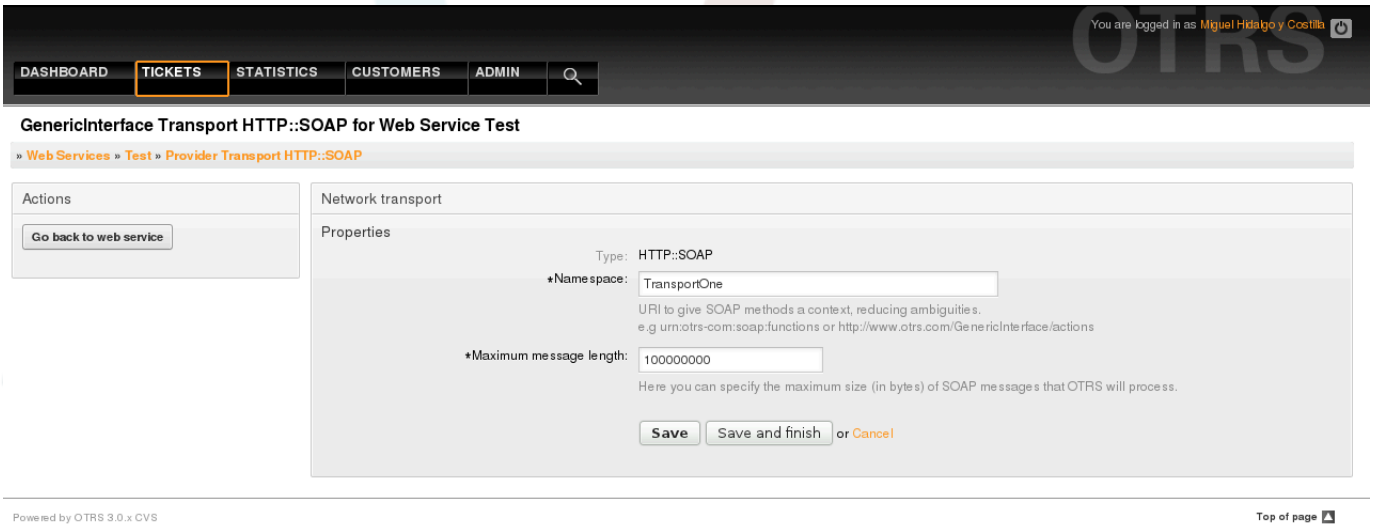


図:ウェブサービス供給者ネットワーク・トランスポート

11.4.3.7.2. ウェブサービス・オペレーション

あなたが供給者としてOTRSを使用している場合、行なうことができるアクションは「オペレーション」と呼ばれます。各オペレーションはコントローラーに所属しています。コントローラーは、オペレーションまたは依頼アクション(Invokers)の蓄積です。通常、同じコントローラーからのオペレーションは、同様の設定を必要で、同じ構成ダイアログを共有します。しかし各オペレーションは必要なら独立した構成ダイアログを持ちます。

名前、記述、バックエンドおよびマッピングは、通常すべてのオペレーションに現われるフィールドです。オペレーションの特定の要求を満たすために、他の特別のフィールドがデフォルトではない構成ダイアログに現われる場合があります。

通常は、各オペレーションに2つのマッピングする構成セクションがあります。一つは入信データ、もう一つは発信データです。各マッピング指示のために異なるマッピングするタイプ(バックエンド)を選ぶことができます。なぜなら、それらの構成が互いから独立して、さらにオペレーション・バックエンドから独立しているからです。正常で、最も一般的な実行は、両方のケースに (逆

の構成を持つ) 同じマッピングタイプを使うオペレーションです。完全なマッピング構成は、マッピングタイプに依存して個別の画面の中で行われます。

オペレーション・バックエンドは事前に投入されており編集可能ではありません。ウェブサービス編集画面に対するオペレーションを選択する時、このパラメタを見るでしょう。フィールドは参考に過ぎません。

アクション・カラム上の画面の左部でに、オプションがあります:「ウェブサービスに戻る」(最後の保存以来変更をすべて廃棄して)と「デリート」。最後のものをクリックしたら、ダイアログが開き、オペレーションを削除したいかどうか尋ねます。オペレーションとその構成の削除を確認するために「削除」ボタンをクリックします。あるいは、削除ダイアログを閉じるために「キャンセル」をクリックします。

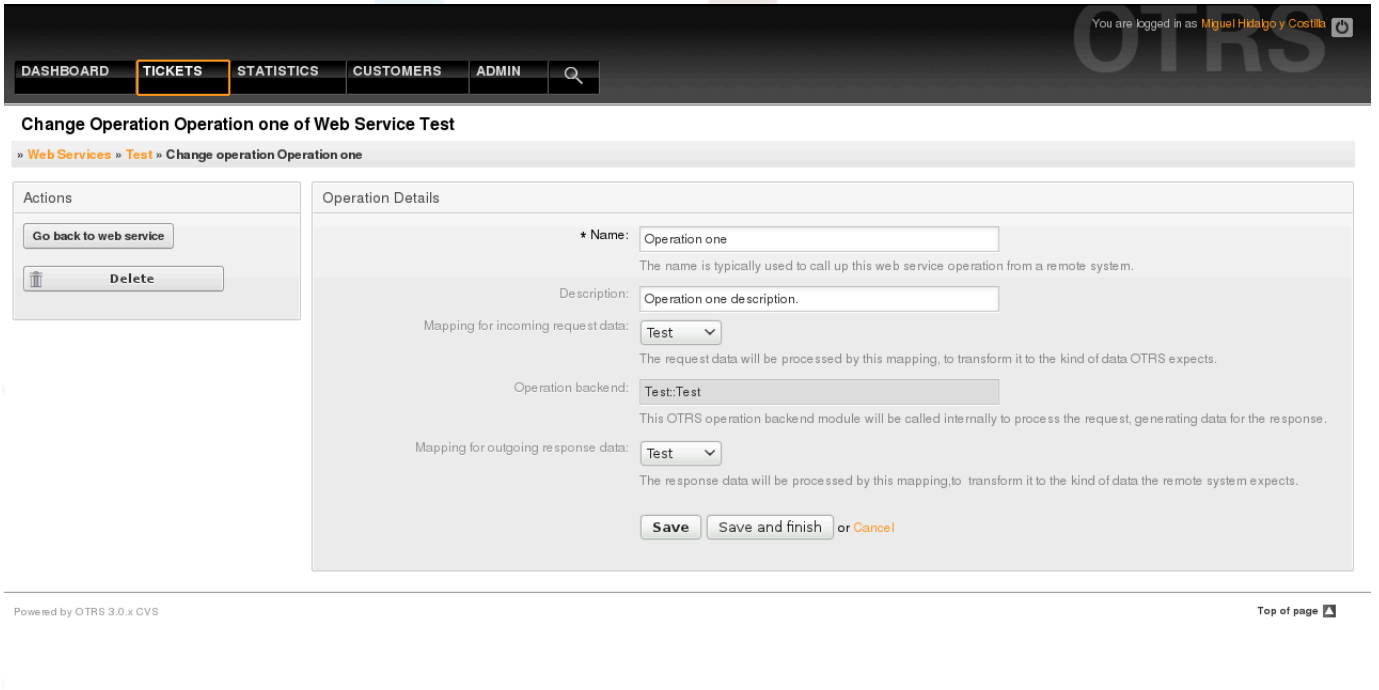


図:ウェブサービス・オペレーション

11.4.3.7.3. ウェブサービス依頼者ネットワーク・トランスポート

依頼者用ネットワークトランスポート構成は供給者用の構成に似ています。依頼者の「HTTP::SOAP」ネットワーク・トランスポートのために、セットすることができるフィールドがさらにもっとあります。

必須フィールドの「エンドポイント」および「ネームスペース」とは別に、(エンドポイントとは、リクエストを受け入れるリモート・システム・ウェブサービス・インターフェースのURIのこと)さらに次のように明示することができます:

- SOAPメッセージのための符号化(utf-8, latin1, iso-8859-1, cp1250など)
- SOAPアクション・ヘッダー:空か埋められたSOAPActionヘッダーを送信するためにこれを使用することができます。「いいえ」をセットすると、SOAPメッセージ上のSOAPアクション・ヘッダは空の文字になります。あるいは　Namespace#Action　フォーマットでSOAPアクションに「はい」をセットし、セパレータを定義します。(典型的なセパレータは　.Netウェブサービスには「/」その他には「#」です。)

- 認証:　認証機構をセットするためには、いかなる認証も使わないか、またはリストからひとつを選択するために、「-」をセットします。そうすればフィールド詳細が現れるでしょう。

注記

現在、「BasicAuth」(HTTP)認証機構だけが実行(impletented)されます。リモート・システム構成に依存して、それを使用するべきかどうか決定することができます。もし使用されれば、リモートシステムにアクセスするためにユーザー・ネームとパスワードを提供しなければなりません。

警告

認証用パスワードを供給しそのあとでYAMLファイルをウェブサービスへエクスポートするならば、このパスワードは暴露され、TYAMLファイルの内部のプレインテキスト文字列に書き込まれるでしょう。必要に応じて予防措置を講じてください。

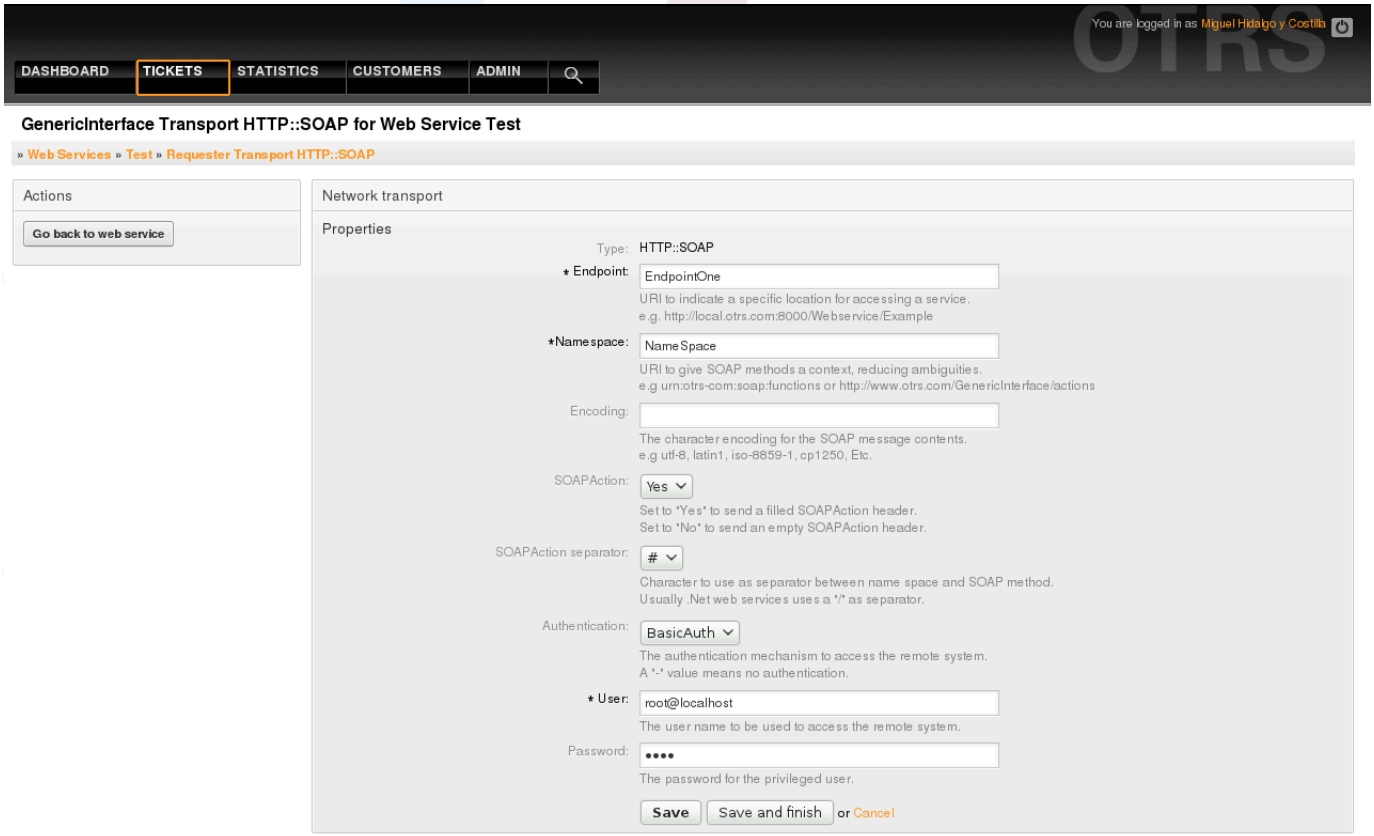


図: ウェブサービス依頼者ネットワーク・トランスポート

11.4.3.7.4. ウェブサービス依頼アクション(Invokers)

あなたが依頼者としてOTRSを使用している場合、行なうことができるアクションは「依頼アクション(Invokers)」と呼ばれます。依頼アクション(Invokers)はそれぞれコントローラーに所属しています。(コントローラーはオペレーションまたは依頼アクション(Invokers)の蓄積である)通常、同じコントローラーからの依頼アクション(Invokers)は、同様のセッティングを必要とし、同じ構成ダイアログを共有します。必要ならば、依頼アクション(Invokers)はそれぞれ独立した構成ダイアログを持つことができます。

名前、記述、バックエンドおよびマッピングはすべての依頼アクション(Invokers)に通常現われるフィールドである。さらにイベントのきっかけと他の特別なフィールドのリストは、依頼アクション(Invokers)の特別な要求を満たすために、デフォルトでない構成ダイアログを現わすことができます。

通常は、各依頼アクション(Invokers)につき2つのマッピング構成セクションがあります。ひとつは入信データでもう一方は発信データです。それらの構成が相互に独立して、さらに依頼アクション(Invokers)バックエンドから独立しているので、各マッピング指示のために異なるマッピングするタイプ(バックエンド)を選ぶことができます。正常で、最も一般的な実行は、依頼アクション(Invokers)が逆の構成で、両方の場合の中で同じマッピングタイプを使用するということです。完全なマッピング構成は個別の画面の中で行われます。それはマッピングタイプに依存します。

依頼アクション(Invokers)バックエンドは事前に投入されており編集可能ではありません。ウェブサービス編集画面に対する依頼アクション(Invokers)を選択する時、このパラメタを見るでしょう。フィールドは参考に過ぎません。

イベント・トリガーは、「TicketCreate」のようなOTRS、「ArticleSend」などの内のイベントです。これらは依頼アクション(Invokers)を実行するためにトリガーとして働くことができます。依頼アクション(Invokers)はそれぞれ少なくとも1つのイベント・トリガーを登録する必要があり、そうでなければ依頼アクション(Invokers)は役立たないでしょう。なぜならそれが呼び出されることがないからです。イベント・トリガーの非同期特性は、OTRSプロセスが依頼アクション(Invokers)を扱うか、あるいはそれがスケジューラに委任されるかどうかを定義します。

注記

OTRSスケジューラは背景のタスクを実行する個別のプロセスである。OTRSプロセスの使用自体はリモートシステムの反応に時間がかかる（利用不能かネットワーク問題がある場合）かどうかには影響されません。ウェブサービスを使用するスケジューラを使用しなければ、OTRSは遅くなりあるいは反応しなくなります。したがって、非同期イベント・トリガーをできるだけしばしば使用することを強く推薦します。

イベント・トリガーを追加するためには、最初に最初のリストからイベント・ファミリーをつぎに第二のリストからイベント名を選択し、それから非同期プロパティをセットします。（イベントのきっかけが非同期ではないという意味で未検証ならば）プラスのボタンを最後にクリックします。新規イベント・トリガーが作成され、それは依頼アクション(Invokers)「イベント・トリガー」リスト上でリストされます。

イベント・トリガーを削除するためには、単に「イベント・トリガー」リスト内で削除したいイベント・トリガーを選択します。そして列末にあるゴミ箱アイコンをクリックします。これによりダイアログが開き、あなたが本当にイベント・トリガーを削除するかを確認します。リストからイベント・トリガーを削除するために「削除」をクリックするか、あるいはダイアログを閉じるために「キャンセル」をクリックして下さい。

アクション・カラム画面の左部には、以下のオプションがあります:「ウェブサービスに戻る」（最後の保存以来変更をすべて廃棄して）と「削除」です。あなたが最後のものをクリックすれば、ダイアログは出現し、あなたが依頼アクション(Invokers)を削除したいかを尋ねます。確認するために「削除」をクリックすると、依頼アクション(Invokers)とその構成の削除する、あるいは「キャンセル」して削除ダイアログを閉じるかを確認されます。

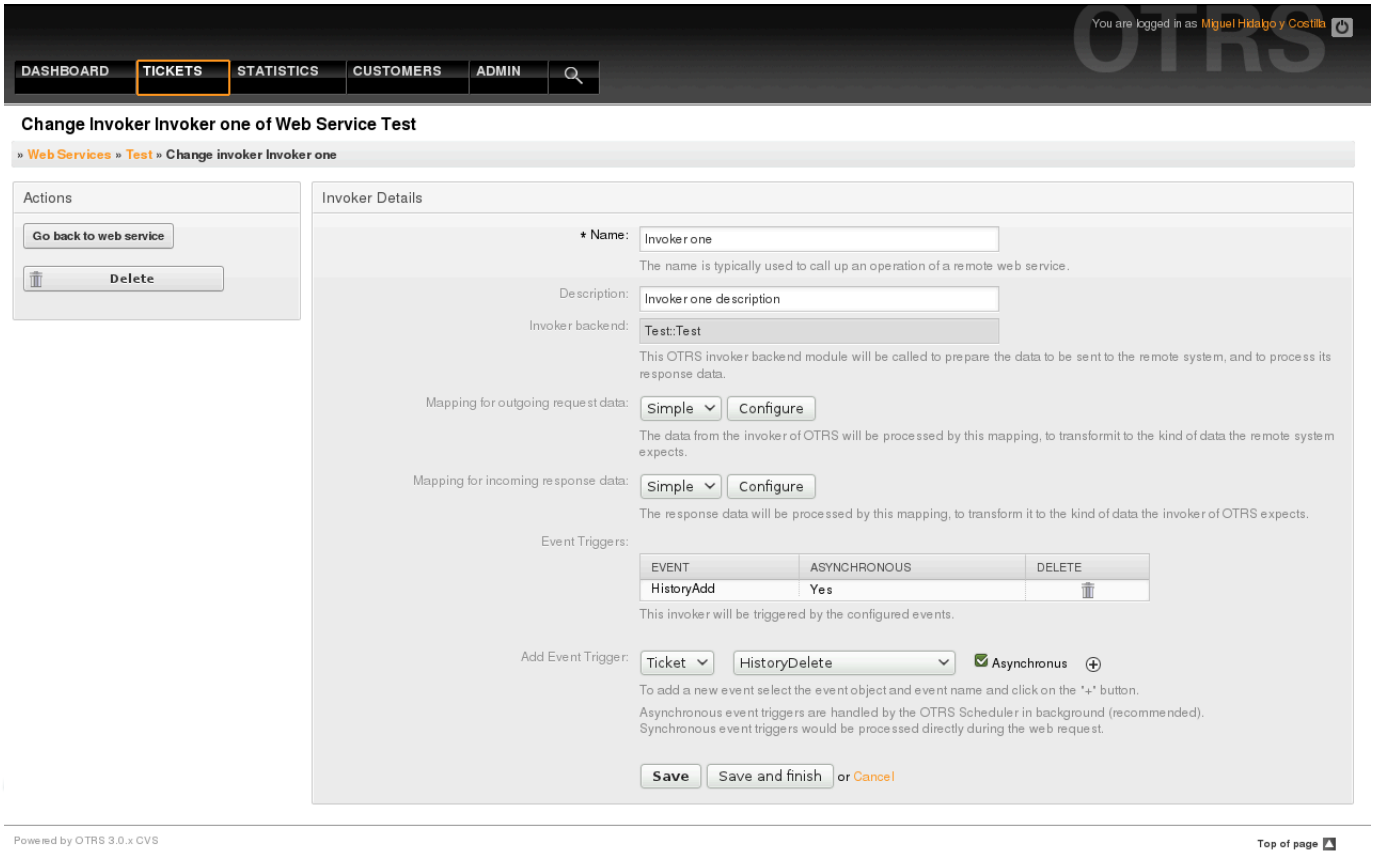


図:ウェブサービス依頼アクション*(Invokers)*

11.4.3.7.5. ウェブサービスマッピング

通常は、ウェブサービスはリモート・システムと相互に作用するために、あるフォーマットから別の(マッピングあるいはデータ構造の変更)フォーマットへデータを変換する必要のある場合があります。別のOTRSシステムでなく、かつ、または、OTRSデータ構造および値を理解できない、といった可能性は非常に高いといえます。これらの場合では、いくらかあるいはすべての値を(そして時々、値(キー)どころか完全な構造の名前さえ)もう1つの端の予期されたデータと一致するために、変更しなければなりません。このタスクを遂行するために、一般インターフェースのマップする層は存在します。

リモート・システムは自身のデータ構造を持ちます。それは新規マッピングモジュールをおのおののケースに対して作成することができます。(例えば、OTRS出荷時で、SAPソリューションマネージャに用にカスタマイズされたマッピング・モジュールがあります。)しかし、それは必ずしも必要だとは限りません。モジュールMapping::Simpleは、ほとんどのマッピング要求をカバーすべきです。

注記

Mapping::Simpleがウェブサービスのすべてのマッピングする必要をカバーするとは限らない場合、新規マッピングモジュールが作成されるべきです。新規マッピングするモジュールを作成する方法に関してもっと学習するために、OTRS開発マニュアルを調べてください。

このモジュールは、デフォルト値をマッピングさせる機会を与えます。各キーあるいは全体のコミュニケーション・データに見合う値のために。

画面の初めに、あなたがデフォルトルールをセットできる一般的なセクションを見ます。そのルールは全てのマップされていないキーや値に適用します。利用可能な3つのオプションがあり、これらのオプションは下のようにリストされます:

- 保持（変更しないままにして）：キーや値をとにかく触りません。
- 無視（キー/値ペアを捨てます):これがキーに適用されれば、それはキーと値を削除します。なぜならキーが削除されると続いてその関連値も削除されるからです。これが値に適用されれば、値だけが削除され、キーは保持されます。この時それは空の値に関連付けられます。
- MapTo（デフォルトとして供給されたキーまたは値の使用）：定義されたマップルールのない全てのキーとまたは値は、デフォルトとしてこれを使います。このオプションを選択するとき、新規のテキストフィールドがこのデフォルトをセットするために現れます。

新規キーマップのために「+」ボタンをクリックすると、単一のマッピング構成用の新規ボックスを表示します。必要なだけ多くのキーマッピングを追加することができます。再度「＋」をクリックするだけで新規マッピングボックスが既存のものの下に現れます。このマッピングボックスから、次のようなオプションを持つ単一のキーのためにマップを定義することができます。

- 正確な値(s):古いキーが正確に一致すれば、古い重要な文字列は新規ものに変更されます。
- 正規表現: 重要なストリングは正規表現規則に従って代替されます。

新規値マップ「+」ボタンを押すと、値マップのための新規列を表示します。ここではキーマップに関して同じオプションでマップすべき各値(正確な値および正規表現)に関するルールを定義することも可能も可能です。必要なだけ多くのマップする値を追加できます。そしてもしそれらのひとつを削除したければ、単に各マッピング値の列の「-」ボタンをクリックします。

完全なキーマッピングセクション（ボックス）を削除可能ならば、単に削除したいボックスの右上端にある「-」ボタンを押します。

完全なマッピング構成を削除する必要がある場合:　　対応するオペレーションか依頼アクション(Invokers)画面に戻って、以前に選択し、その値を「-」にセットするというマッピング指示を捜して、変更を適用するために構成を保存してください。

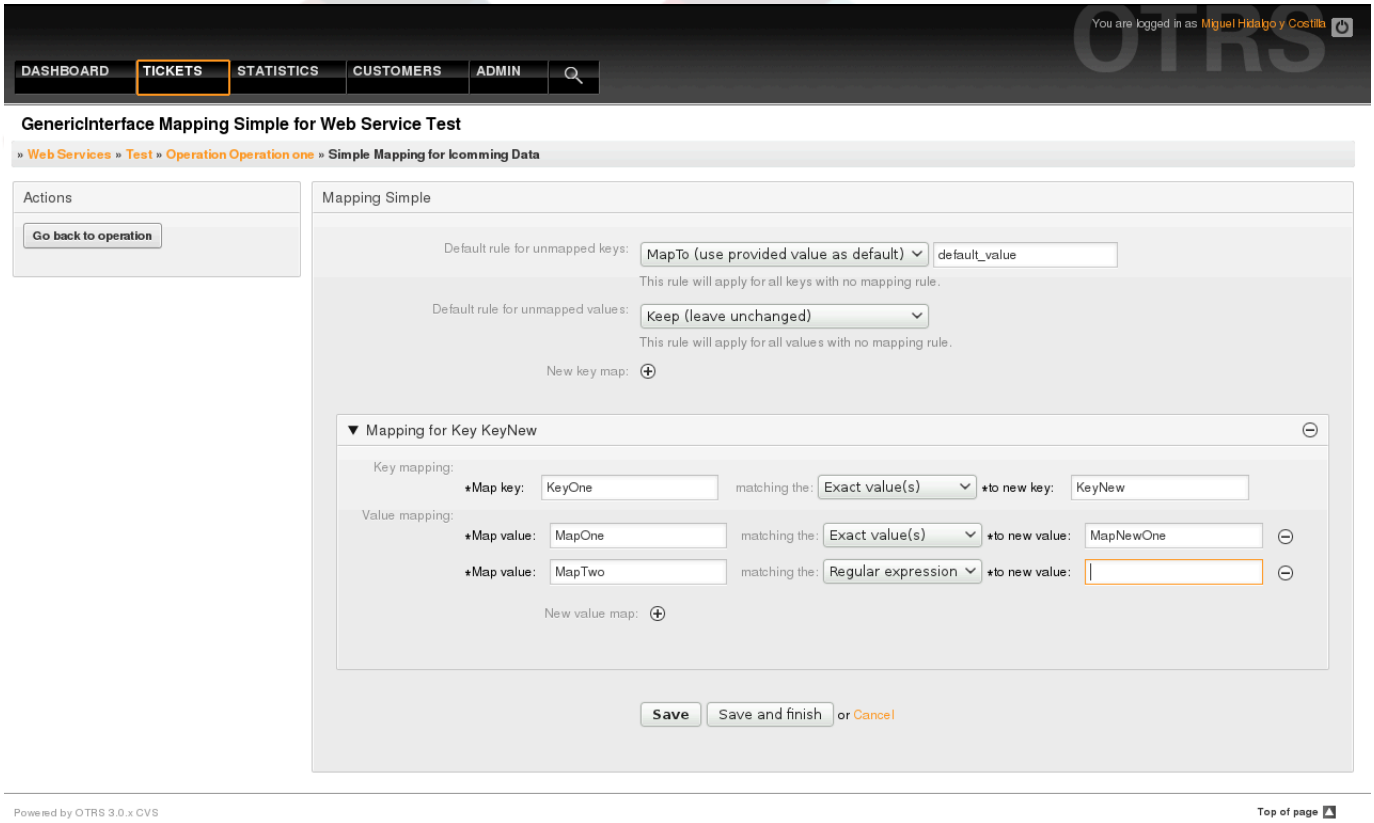


図:ウェブサービスマッピング

11.5. ウェブサービス・コマンドライン・インターフェイス

コマンドライン・インタフェース(CLI)はウェブサービスとジョブをする速い方法です。それは、基本操作を行なうために使用できるツールセットから成ります:

- YAMLファイルに基づいてウェブサービスを作成、更新、読み込み、リスト、削除します。
- フィルタオプションでデバッガログを読み込みます。

> **注記**
>
> ウェブサービスと連動するするためにCLIを使用する必要はありません。管理者インターフェースへ統合された、ウェブサービスのすべての部分と相互作用する画面の完全セットがあります。このマニュアルに含まれたウェブサービスGUIセクションを読んでください。

11.5.1. ウェブサービス構成

「WebserviceConfig.pl」は基本的だが、速く、強力です、ウェブサービス構成で働くツールとして開発されました。次のアクションを実行できるようになります:

- 追加：構成ソースとしてYAMLファイルを使うウェブサービスを作成します。
- 更新：既存のウェブサービスを変更します。異なるまたは変更されたYAMLファイルを使って構成は変更可能です。
- 読み込み：画面に表示されたウェブサービスの構成を入手します。
- リスト:システムに登録されたですべてのウェブサービスの完全なリストを入手します。
- 削除:システムからウェブサービスを削除すること。このアクションは取消不能なので利用の際はご注意ください。

> **警告**
>
> ウェブサービスREADオペレーションは、プレインテキストとして画面にすべての構成を表示します。これには格納されたパスワードを含みます。これに注意して、必要とされる予防措置を講じてください!

サンプル：新規ウェブサービス構成の作成:

```
                shell> OTRS_HOME/bin/otrs.WebserviceConfig.pl -a write -n <webservice_name>
-f /path/to/yaml/file
```

さらに、次のオプションを備えた「otrs.WebserviceConfig.pl」を使用することができます:

- **-a read -i <webservice_id>** -　格納された構成を読むこと。
- **-a write -n <webservice_name> -f /path/to/yaml/file** - 新規ウェブサービスを作ること。
- **-a write -i <webservice_id> -f /path/to/yaml/file** - ウェブサービスを更新すること。
- **-a list** - 利用可能なウェブサービスをリストすること。

- **-a delete -i <webservice_id>** - ウェブサービスを削除すること。

11.5.2. ウェブサービス・デバッガ

コマンドライン・インタフェースによる別の利用可能なツール
は"otrs.GenericInterfaceDebugRead.pl"　スクリプトです。それはウェブサービス・デバッガ・ログエントリを検索するインターフェイスです。

サンプル: デバッガ・ログエントリの検索:

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

オプションのパラメタは"otrs.GenericInterfaceDebugRead.pl"のために使用されます。

- **-c** - コミュニケーションID(md5sumフォーマット)によってフィルタします。
- **-t** - CommunicationType(「プロバイダー」あるいは「依頼者」)によってフィルタします。
- **-a** - 日付(その日、またはある日の後)によってフィルタします。
- **-b** - 日付(その日、またはある日の後)によってフィルタします。
- **-i** - IPアドレス(有効なIPv4かIPv6アドレス)によってフィルタします。
- **-w** - ウェブサービスIDによってフィルタします。
- **-d** - 詳細なコミュニケーション・データを含みます。

サンプル: すべてのパラメタを備えたデバッガ・ログエントリを検索:

```
shell> ./otrs.GenericInterfaceDebugRead.pl -c
a7cc4d9f5c70387a9bfbe1351bc88966 -t Provider -a '2011-07-22 00:00:00' -b '2011-07-26
00:00:00' -i 127.0.0.1 -w 123 -d 1
```

注記

上記にリストされたフィルタ・オプションの少なくともひとつを含んでいること強くお薦めします、"-d"オプションが選択されていればさらに良いです。　なぜならば多くの情報はデータベースから引き出され画面に表示され、さらに実際より必要以上の多くの情報が遅いレスポンス時間の原因となりえます。

11.6. ウェブサービス構成

その設計から、ウェブサービスは
　単一のOTRSシステムから別のシステム (例えば、テストか開発環境からプロダクション・システムまで) まで
　ポータブルになるように着想されました。したがって、それはデータベースからウェブサービス構成を抽出し、かつそれを別のものにインポートする、容易な方法を持つために必要でした。このタスクを遂行するために、一般インターフェースはウェブサービス構成基礎としてYAMLファイルを使用します。

なぜYAMLなのでしょう?
　YAMLは人間が読み書きするのに優しいことを目指したマークアップ言語です(JSONより理解するほうが簡単です)。YAMLは数的タグのようなXMLが持ついくつかの制限を持ちません。それはオープンで標準化されており、全ウェブサービス構成を格納するほどに十分に完全です。

注記

YAMLについてより知りたければ http://www.yaml.org/ をご覧ください。

下記はYAMLフォーマットによるウェブサービス設定ファイル例です:

```
---
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. 構成詳細

11.6.1.1. 一般

- Description(摘要):ウェブサービスを記述する短いテキスト。
- RemoteSystem（リモート・システム）:リモート・システムの短い摘要。
- Debugger(デバッガ）:デバッガ・セッティング用のコンテナ。
- Provider(供給者):供給者セッティング用のコンテナ。
- Requester(依頼者):依頼者セッティング用のコンテナ。

11.6.1.2. デバッガ

- デバッグ閾値：デバッガ・レベル

  可能な値

  - debug(デバッグ): 全てのログはすべてデータベースに格納されます。
  - info(情報): 情報、通知およびエラーレベルログはデータベースに格納されます。
  - notice(通知): 通知、エラーレベルログデータベースに格納されます。
  - error(エラー): エラーレベルログだけがデータベースに格納されます。

11.6.1.3. 供給者

- Operation(オペレーション): 各オペレーション設定用コンテナ。
- Transport（トランスポート）: ネットワーク・トランスポート用コンテナ。

11.6.1.3.1. オペレーション

- <OperationName>: オペレーション用のユニークな名前、それ自身のオペレーション設定用コンテナ(基数0..n, しかし重複しない).

11.6.1.3.1.1. <OperationName>

このセクションは、他のオペレーションがもっと含んでいるかもしれないタイ「Test::Test」、あるいは異なるセッティング、からのオペレーションに基づきます。

- Description (摘要) :オペレーションを記述する短いテキスト。
- MappingInbound (マッピング・インバウンド) : 入信依頼データ用マッピング設定用コンテナ。
- MappingOutbound (マッピング・アウトバウンド) : 発信応答データ用マッピング用設定コンテナ。
- Type (タイプ) : Controller::Operation formatでのオペレーション・バックエンド。

11.6.1.3.1.1.1. MappingInbound (マッピング・インバウンド)

このセクションは、タイプ「Simple」からのマッピングに基づきます。他のマッピングはより多くの異なる設定を含みます。

- Config(構成): このマッピング設定用コンテナ。
- Type (タイプ) : マッピング・バックエンド。

11.6.1.3.1.1.1.1. Config (構成)

- KeyMapDefault: マップ未設定キーのコンテナ
- ValueMapDefault: 全てのマップ未設定の値の設定のコンテナ。
- KeyMapExact: 全ての正確なキーマッピング (基数 0 .. 1)。
- KeyMapRegEx: 全ての正規表現キーマッピング用コンテナ (基数 0 .. 1)。
- ValueMap: 全ての値マッピング用コンテナ (基数 0 .. 1)。

11.6.1.3.1.1.1.1.1. KeyMapDefault (マップ未設定キーのコンテナ)

- MapTo: 使用される新しい値 (MapTypeがMapToにセットされた場合のみ適用可能)。
- MapType: マッピング用規則。

  可能な値

  - Keep (保持) : 変更なく終了。
  - Ignore (無視) : やめる
  - MapTo (マップトゥー) : MapTo値を変更

11.6.1.3.1.1.1.1.2. ValueMapDefault (値マップ・デフォルト)

KeyMapDefault (キーマップ・デフォルト) に似ています。

11.6.1.3.1.1.1.1.3. KeyMapExact (正確なキーマップ)

- <oldkey>: <newkey> (基数 0 .. n 但し重複なく)。

11.6.1.3.1.1.1.1.4. KeyMapRegEx

- <oldkey(RegEx)>: <newkey> (基数 0 .. n 但し重複なく)。

11.6.1.3.1.1.1.1.5. ValueMap

- <newkey>: a container for value mappings for this new key (KeyMapExact and KeyMapRegExからの新規キーに依存する基数)

11.6.1.3.1.1.1.1.5.1. <newkey>

- ValueMapExact: 全ての性格な値マッピング用コンテナ (基数 0 .. 1)。
- ValueMapRegEx: 全ての正規表現値のマッピング用コンテナ(基数 0 .. 1)。

11.6.1.3.1.1.1.1.5.1.1. valueMapExact

- <oldvalue>: <newvalue> (基数 0 .. n 但し重複なく)。

11.6.1.3.1.1.1.1.5.1.2. ValueMapRegEx

- <oldvalue(RegEx)>: <newvalue> (基数 0 .. n 但し重複なく)。

11.6.1.3.1.1.2. MappingOutbound

MappingInboundと同じ。

11.6.1.3.1.1.3. トランスポート

このセクションは、供給者ネットワーク・トランスポートHTTP::SOAPに基づきます。他のトランスポートはより多く異なるセッティングを含んでいるかもしれません。

- Config (構成) : 特定のネットワークトランスポート構成セッティング用のコンテナー。
- Type: 供給者ネットワークトランスポートバックエンド。

11.6.1.3.1.1.3.1. Config (構成)

- MaxLength: OTRSによってSOAPメッセージで読まれるバイトで最大の長さ。
- NameSpace:　すべてのオペレーションにこのウェブサービスが所有しているコンテキストを与えるURI。

11.6.1.4. 依頼者

- Invoker (依頼アクション) : 各依頼アクション(Invokers)のセッティング用のコンテナ。
- Transport(トランスポート): 依頼者ネットワーク・トランスポート設定用コンテナ。

11.6.1.4.1. 依頼アクション(Invokers)

- <InvokerName>: 依頼アクション用のユニーク名前。依頼アクションそれ自身の設定用コンテナ。(基数0..n 但し重複なく)

11.6.1.4.1.1. <InvokerName>

このセクションは、タイプ"Test::Test"からの依頼アクションに基づいています。他の依頼アクションはより多く異なる設定を含んでいるかもしれません。

- Description (摘要) : 依頼アクションを記述する短いテキスト
- Events (イベント) : イベント・トリガー設定の未命名のリスト用コンテナ
- MappingInbound (マッピング・インバウンド) : 入信応答データ用マッピング設定用コンテナ
- MappingOutbound (マッピング・アウトバウンド) : 発信依頼データ用マッピング設定用コンテナ

- Type (タイプ) : Controller::Invoker フォーマットでの依頼アクションのバックエンド

11.6.1.4.1.1.1. イベント

- リスト要素: (基数 0 .. n)
  - Asynchronous (非同期) : もし依頼アクションの実行がスケジューラに可能な値を委任するならセットします。

    ##### 可能な値

    - 0: スケジューラで取扱い不能です。
    - 1: スケジューラで取扱い可能です。
  - Event (イベント) : イベント・。トリガー名。

    ##### **Possible Values** (チケットのイベント用の)

    - チケット作成
    - TicketDelete (チケット削除)
    - TicketTitleUpdate (チケット・タイトル・更新)
    - TicketUnlockTimeoutUpdate (チケット・アンロック・タイムアウト更新)
    - TicketQueueUpdate (チケットキュー更新)
    - TicketTypeUpdate (チケット・タイプ・更新)
    - TicketServiceUpdate (チケットサービス更新)
    - TicketSLAUpdate (チケットSLA更新)
    - TicketCustomerUpdate (チケット顧客更新)
    - TicketFreeTextUpdate (チケットフリーテキスト更新)
    - TicketFreeTimeUpdate (チケットフリータイム更新)
    - TicketPendingTimeUpdate (チケット保留時間更新)
    - TicketLockUpdate (チケットロック更新)
    - TicketArchiveFlagUpdate (チケットアーカイブ更新)
    - TicketStateUpdate (チケット・状態・更新)
    - TicketOwnerUpdate (チケット所有者更新)
    - TicketResponsibleUpdate (チケット応答可能更新)
    - TicketPriorityUpdate (チケット優先権更新)
    - HistoryAdd (履歴追加)
    - HistoryDelete (履歴削除)
    - TicketAccountTime (チケットアカウント時間)
    - TicketMerge (チケット結合)

- TicketSubscribe (チケット・登録)
- TicketUnsubscribe (チケット・登録解除)
- TicketFlagSet (チケットフラグセット)
- TicketFlagDelete (チケットフラグ削除)
- TicketSlaveLinkAdd (チケットスレーブリンク追加)
- TicketSlaveLinkDelete (チケットスレーブリンク削除)
- TicketMasterLinkDelete (チケットマスターリンク削除)

Possible Values (for article events) (可能な値 (記事イベントに関する))

- Article Events (記事イベント)
- ArticleCreate (記事作成)
- ArticleFreeTextUpdate (記事フリーテキスト更新)
- ArticleUpdate (記事更新)
- ArticleSend (記事送信)
- ArticleBounce (記事返答)
- ArticleAgentNotification (記事担当者通知)
- ArticleCustomerNotification (記事顧客通知)
- ArticleAutoResponse (記事自動返信)
- ArticleFlagSet (記事フラグセット)
- ArticleFlagDelete (記事フラグ削除)
- ArticleAgentNotification (記事担当者通知)
- ArticleCustomerNotification (記事顧客通知)

11.6.1.4.1.1.2. MappingInbound (マッピング・インバウンド)

オペレーションMappingInboundと同じ

11.6.1.4.1.1.3. MappingOutbound

オペレーションMappingInboundと同じ。

11.6.1.4.1.1.4. トランスポート

このセクションは、依頼者ネットワーク・トランスポートHTTP::SOAPに基づきます。他のトランスポートはより多く異なる設定を含むかもしれません。

- Config (構成) : 特定のネットワークトランスポート構成セッティング用のコンテナー。
- Type (タイプ) : 依頼者ネットワークトランスポートバックエンド

11.6.1.4.1.1.4.1. Config (構成)

- Authentication(認証): 認証セッティング用のコンテナ

- Encoding(エンコーディング): SOAPメッセージ・リクエスト符号化
- Endpoint(エンドポイント): OTRSリクエストを受け入れるリモート・サーバ・ウェブサービスのURI
- NameSpace(名前空間): すべての依頼アクション(Invokers)にこのウェブサービスが所有しているコンテキストを与えるURI.
- SOAPAction(SOAPアクション): SOAPメッセージ(in "<NameSpace> <Separator> <Action>" format).の中の空か満たされたSOAPActionヘッダーを送ること。

  可能な値

  - YES(はい): 埋まったSOAPActionヘッダーを送る。
  - No(いいえ): 空のSOAPActionヘッダーを送る
- SOAPActionSeparator: 記入された SOAPAction ヘッダの <Separator> をセットする。

  可能な値

  - '/': .netウェブサービスのために使用されます。
  - '#': すべての休息ウェブサービスのために使用されます。

11.6.1.4.1.1.4.1.1. 認証

- User (ユーザ) : リモート・ウェブサービスにアクセスする特権ユーザー名
- Password (パスワード) : プレーンテキスト中の特権ユーザー用のパスワード
- Type (タイプ) : 認証のタイプ

11.7. コネクタ

コネクタは、OTRSがウェブサービス供給者あるいは依頼アクションとして働く場合、あるいは、OTRSがウェブサービス依頼者として働く場合、オペレーションとも呼ばれるアクションのセットの核心になります。しかし、さらに、それは特別のマッピングあるいはトランスポートも含むことができます。

1つのコネクタは、オペレーションのみ、依頼アクション(Invokers)のみ、あるいはその両方だけ、であり得ます。コネクタはマッピングやトランスポートのように他のコネクタのパーツを利用することさえ可能です。そのようなことは何もコネクタにとっては特殊なことではなくむしろそれらを実装しようとしている。

言いかえれば、コネクタは単なるコントローラー層に制限されていない。しかし、もし必要ならばそれはデータマッピングまたはネットワークのトランスポート層まで拡張することができます。

一般インターフェースのモジュール設計により、コネクタはプラグ・インと見なすことができます。;これは、コネクタを加えることによって、次のものを使用して、一般インターフェースの能力を拡張することができることを意味します: 使用法：OTRS機能追加、OTRSカスタム・モジュール、サードパーティ・モジュール等。

11.7.1. バンドル・コネクタ

OTRSのこのバージョンで含まれて、次のコネクタは、使用される準備ができています。

- セッション
- チケット

11.7.1.1. セッション・コネクタ

このコネクタは他のオペレーションの中で使用することができる有効なSessionIDを作成するのに有能である。

提供：

- オペレーション:
  - セッション作成

11.7.1.1.1. オペレーション

11.7.1.1.1.1. セッション作成

SessionCreateは、TicketCreateのような他のコネクタからの他のオペレーションの中で使用される、新規新規有効なSessionIDを作成します。

> **注記**
>
> 他のコネクタからの他のオペレーションの中でSessionIDを使用するために、オペレーションはSessionIDによって認証を実装する必要がある。バンドルされたオペレーションの残りはすべて認証方法として有効なSessionIDを受け入れることができます。

可能な属性:

```
<SessionCreate>
   <!--このレベルでは次の 2 つの必須選択項目があります。-->
   <!--Optional:-->
   <UserLogin>?</UserLogin>
   <!--Optional:-->
   <CustomerUserLogin>?</CustomerUserLogin>
   <!--Optional:-->
   <Password>?</Password>
</SessionCreate>
```

11.7.1.2. チケット・コネクタ

このコネクタは、チケットとダイアログするために基本機能を供給します。

提供：

- オペレーション:
  - チケット作成
  - TicketCreate(チケット作成)
  - TicketGet(チケット取得)
  - TicketSearch(チケット検索)

11.7.1.2.1. オペレーション

11.7.1.2.1.1. チケット作成

TicketCreateは、OTRSにチケットを作成するインターフェースを提供します。チケットは記事を含んでいなければいけないし、いくつか添付を含むことができます。全ての定義された動的フィールドも、TicketCreateオペレーション上でセットすることができます。

可能な属性:

```
<TicketCreate>
   <!--このレベルでは次の 3 つの必須選択項目があります。-->
   <!--Optional:-->
   <UserLogin>?</UserLogin>
   <!--Optional:-->
   <CustomerUserLogin>?</CustomerUserLogin>
   <!--Optional:-->
   <SessionID>?</SessionID>
   <!--Optional:-->
   <Password>?</Password>
   <Ticket>
      <Title>?</Title>
      <!--このレベルでは次の 2 つの必須選択項目があります。-->
      <!--Optional:-->
      <QueueID>?</QueueID>
      <!--Optional:-->
      <Queue>?</Queue>
      <!--このレベルでは次の 2 つの選択項目があります。-->
      <!--Optional:-->
      <TypeID>?</TypeID>
      <!--Optional:-->
      <Type>?</Type>
      <!--このレベルでは次の 2 つの選択項目があります。-->
      <!--Optional:-->
      <ServiceID>?</ServiceID>
      <!--Optional:-->
      <Service>?</Service>
      <!--このレベルでは次の 2 つの選択項目があります。-->
      <!--Optional:-->
      <SLAID>?</SLAID>
      <!--Optional:-->
      <SLA>?</SLA>
      <!--このレベルでは次の3つの必須選択項目があります。-->
      <!--Optional:-->
      <StateID>?</StateID>
      <!--Optional:-->
      <State>?</State>
      <!--このレベルでは次の 2 つの必須選択項目があります。-->
      <!--Optional:-->
      <PriorityID>?</PriorityID>
      <!--Optional:-->
      <Priority>?</Priority>
      <!--このレベルでは次の 2 つの選択項目があります。-->
      <!--Optional:-->
      <OwnerID>?</OwnerID>
      <!--Optional:-->
      <Owner>?</Owner>
      <!--このレベルでは次の 2 つの選択項目があります。-->
      <!--Optional:-->
      <ResponsibleID>?</ResponsibleID>
      <!--Optional:-->
      <Responsible>?</Responsible>
      <CustomerUser>?</CustomerUser>
      <!--Optional:-->
      <CustomerID>?</CustomerID>
      <!--Optional:-->
      <PendingTime>
         <Year>?</Year>
         <Month>?</Month>
         <Day>?</Day>
         <Hour>?</Hour>
         <Minute>?</Minute>
      </PendingTime>
   </Ticket>
   <Article>
      <!--このレベルでは次の 2 つの選択項目があります。-->
      <!--Optional:-->
      <ArticleTypeID>?</ArticleTypeID>
```

```
            <!--Optional:-->
            <ArticleType>?</ArticleType>
            <!--このレベルでは次の 2 つの選択項目があります。-->
            <!--Optional:-->
            <SenderTypeID>?</SenderTypeID>
            <!--Optional:-->
            <SenderType>?</SenderType>
            <!--Optional:-->
            <From>?</From>
            <Subject>?</Subject>
            <Body>?</Body>
            <!--このレベルでは次の 2 つの選択項目があります。-->
            <!--Optional:-->
            <ContentType>?</ContentType>
            <Charset>?</Charset>
            <MimeType>?</MimeType>
            <!--Optional:-->
            <HistoryType>?</HistoryType>
            <!--Optional:-->
            <HistoryComment>?</HistoryComment>
            <!--Optional:-->
            <AutoResponseType>?</AutoResponseType>
            <!--Optional:-->
            <TimeUnit>?</TimeUnit>
            <!--Optional:-->
            <NoAgentNotify>?</NoAgentNotify>
            <!--0 回以上の繰り返し:-->
            <ForceNotificationToUserID>?</ForceNotificationToUserID>
            <!--0 回以上の繰り返し:-->
            <ExcludeNotificationToUserID>?</ExcludeNotificationToUserID>
            <!--0 回以上の繰り返し:-->
            <ExcludeMuteNotificationToUserID>?</ExcludeMuteNotificationToUserID>
         </Article>
         <!--0 回以上の繰り返し:-->
         <DynamicField>
            <Name>?</Name>
            <!--1回以上の繰り返し:-->
            <Value>?</Value>
         </DynamicField>
         <!--0 回以上の繰り返し:-->
         <Attachment>
            <Content>cid:61886944659</Content>
            <ContentType>?</ContentType>
            <Filename>?</Filename>
         </Attachment>
      </TicketCreate>
```

11.7.1.2.1.2. TicketCreate(チケット作成)

チケット更新オペレーションは、既存のチケットからの属性を修正するかあるいは新規記事を追加する能力を追加します。その記事は、チケットと新規記事のための添付と全ての定義済動的フィールドを含みます。

注記

チケット属性を修正する新規記事を作成することは必要ではありません。

可能な属性:

```
      <TicketUpdate>
         <!--このレベルでは次の 3 つの必須選択項目があります。-->
         <!--Optional:-->
         <UserLogin>?</UserLogin>
         <!--Optional:-->
         <CustomerUserLogin>?</CustomerUserLogin>
         <!--Optional:-->
         <SessionID>?</SessionID>
```

```
         <!--Optional:-->
         <Password>?</Password>
         <!--このレベルでは次の 2 つの選択項目があります。-->
         <TicketID>?</TicketID>
         <TicketNumber>?</TicketNumber>
         <!--Optional:-->
         <Ticket>
            <!--Optional:-->
            <Title>?</Title>
            <!--このレベルでは次の 2 つの選択項目があります。-->
            <!--Optional:-->
            <QueueID>?</QueueID>
            <!--Optional:-->
            <Queue>?</Queue>
            <!--このレベルでは次の 2 つの選択項目があります。-->
            <!--Optional:-->
            <TypeID>?</TypeID>
            <!--Optional:-->
            <Type>?</Type>
            <!--このレベルでは次の 2 つの選択項目があります。-->
            <!--Optional:-->
            <ServiceID>?</ServiceID>
            <!--Optional:-->
            <Service>?</Service>
            <!--このレベルでは次の 2 つの選択項目があります。-->
            <!--Optional:-->
            <SLAID>?</SLAID>
            <!--Optional:-->
            <SLA>?</SLA>
            <!--このレベルでは次の 2 つの選択項目があります。-->
            <!--Optional:-->
            <StateID>?</StateID>
            <!--Optional:-->
            <State>?</State>
            <!--このレベルでは次の 2 つの選択項目があります。-->
            <!--Optional:-->
            <PriorityID>?</PriorityID>
            <!--Optional:-->
            <Priority>?</Priority>
            <!--このレベルでは次の 2 つの選択項目があります。-->
            <!--Optional:-->
            <OwnerID>?</OwnerID>
            <!--Optional:-->
            <Owner>?</Owner>
            <!--このレベルでは次の 2 つの選択項目があります。-->
            <!--Optional:-->
            <ResponsibleID>?</ResponsibleID>
            <!--Optional:-->
            <Responsible>?</Responsible>
            <!--Optional:-->
            <CustomerUser>?</CustomerUser>
            <!--Optional:-->
            <CustomerID>?</CustomerID>
            <!--Optional:-->
            <PendingTime>
               <Year>?</Year>
               <Month>?</Month>
               <Day>?</Day>
               <Hour>?</Hour>
               <Minute>?</Minute>
            </PendingTime>
         </Ticket>
         <!--Optional:-->
         <Article>
            <!--このレベルでは次の 2 つの選択項目があります。-->
            <!--Optional:-->
            <ArticleTypeID>?</ArticleTypeID>
            <!--Optional:-->
            <ArticleType>?</ArticleType>
            <!--このレベルでは次の 2 つの選択項目があります。-->
            <!--Optional:-->
            <SenderTypeID>?</SenderTypeID>
```

```
            <!--Optional:-->
            <SenderType>?</SenderType>
            <!--Optional:-->
            <From>?</From>
            <Subject>?</Subject>
            <Body>?</Body>
            <!--このレベルでは次の 2 つの選択項目があります。-->
            <!--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(チケット取得)

このオペレーションは全てのチケットの属性を得るために使用されます。(この属性は、動的フィールド、すべての記事、およびチケットに属する全ての添付を含みます。)

可能な属性:

```
      <TicketGet>
         <!--このレベルでは次の3つの必須選択項目があります。-->
         <!--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(チケット検索)

TicketSearchオペレーションは、事前定義された基準に一致するチケットIDのリストを返します。

可能な属性:

```
   <TicketSearch>
      <!--このレベルでは次の3つの必須選択項目があります。-->
      <!--Optional:-->
      <UserLogin>?</UserLogin>
      <!--Optional:-->
      <CustomerUserLogin>?</CustomerUserLogin>
      <!--Optional:-->
      <SessionID>?</SessionID>
      <!--Optional:-->
      <Password>?</Password>
      <!--Optional:-->
      <Limit>?</Limit>
      <!--0回以上繰り返し:-->
      <TicketNumber>?</TicketNumber>
      <!--0回以上繰り返し:-->
      <Title>?</Title>
      <!--0回以上繰り返し:-->
      <Queues>?</Queues>
      <!--0回以上繰り返し:-->
      <QueueIDs>?</QueueIDs>
      <!--Optional:-->
      <UseSubQueues>?</UseSubQueues>
      <!--0回以上繰り返し:-->
      <Types>?</Types>
      <!--0回以上繰り返し:-->
      <TypeIDs>?</TypeIDs>
      <!--0回以上繰り返し:-->
      <States>?</States>
      <!--0回以上繰り返し:-->
      <StateIDs>?</StateIDs>
      <!--0回以上繰り返し:-->
      <StateType>?</StateType>
      <!--0回以上繰り返し:-->
      <StateTypeIDs>?</StateTypeIDs>
      <!--0回以上繰り返し:-->
      <Priorities>?</Priorities>
      <!--0回以上繰り返し:-->
      <PriorityIDs>?</PriorityIDs>
      <!--0回以上繰り返し:-->
      <Services>?</Services>
      <!--0回以上繰り返し:-->
      <ServiceIDs>?</ServiceIDs>
      <!--0回以上繰り返し:-->
      <SLAs>?</SLAs>
      <!--0回以上繰り返し:-->
      <SLAIDs>?</SLAIDs>
      <!--0回以上繰り返し:-->
      <Locks>?</Locks>
      <!--0回以上繰り返し:-->
      <LockIDs>?</LockIDs>
      <!--0回以上繰り返し:-->
      <OwnerIDs>?</OwnerIDs>
```

```
         <!--0回以上繰り返し:-->
         <ResponsibleIDs>?</ResponsibleIDs>
         <!--0回以上繰り返し:-->
         <WatchUserIDs>?</WatchUserIDs>
         <!--0回以上繰り返し:-->
         <CustomerID>?</CustomerID>
         <!--0回以上繰り返し:-->
         <CustomerUserLogin>?</CustomerUserLogin>
         <!--0回以上繰り返し:-->
         <CreatedUserIDs>?</CreatedUserIDs>
         <!--0回以上繰り返し:-->
         <CreatedTypes>?</CreatedTypes>
         <!--0回以上繰り返し:-->
         <CreatedTypeIDs>?</CreatedTypeIDs>
         <!--0回以上繰り返し:-->
         <CreatedPriorities>?</CreatedPriorities>
         <!--0回以上繰り返し:-->
         <CreatedPriorityIDs>?</CreatedPriorityIDs>
         <!--0回以上繰り返し:-->
         <CreatedStates>?</CreatedStates>
         <!--0回以上繰り返し:-->
         <CreatedStateIDs>?</CreatedStateIDs>
         <!--0回以上繰り返し:-->
         <CreatedQueues>?</CreatedQueues>
         <!--0回以上繰り返し:-->
         <CreatedQueueIDs>?</CreatedQueueIDs>
         <!--0回以上繰り返し:-->
         <DynamicFields>
            <!--このレベルでは次の6つの必須選択項目があります。-->
            <!--Optional:-->
            <Equals>?</Equals>
            <!--Optional:-->
            <Like>?</Like>
            <!--Optional:-->
            <GreaterThan>?</GreaterThan>
            <!--Optional:-->
            <GreaterThanEquals>?</GreaterThanEquals>
            <!--Optional:-->
            <SmallerThan>?</SmallerThan>
            <!--Optional:-->
            <SmallerThanEquals>?</SmallerThanEquals>
         </DynamicFields>
         <!--Optional:-->
         <Ticketflag>
            <!--Optional:-->
            <Seen>?</Seen>
         </Ticketflag>
         <!--Optional:-->
         <From>?</From>
         <!--Optional:-->
         <To>?</To>
         <!--Optional:-->
         <Cc>?</Cc>
         <!--Optional:-->
         <Subject>?</Subject>
         <!--Optional:-->
         <Body>?</Body>
         <!--Optional:-->
         <FullTextIndex>?</FullTextIndex>
         <!--Optional:-->
         <ContentSearch>?</ContentSearch>
         <!--Optional:-->
         <ConditionInline>?</ConditionInline>
         <!--Optional:-->
         <ArticleCreateTimeOlderMinutes>?</ArticleCreateTimeOlderMinutes>
         <!--Optional:-->
         <ArticleCreateTimeNewerMinutes>?</ArticleCreateTimeNewerMinutes>
         <!--Optional:-->
         <ArticleCreateTimeNewerDate>?</ArticleCreateTimeNewerDate>
         <!--Optional:-->
         <ArticleCreateTimeOlderDate>?</ArticleCreateTimeOlderDate>
         <!--Optional:-->
```

```
        <TicketCreateTimeOlderMinutes>?</TicketCreateTimeOlderMinutes>
        <!--Optional:-->
        <ATicketCreateTimeNewerMinutes>?</ATicketCreateTimeNewerMinutes>
        <!--Optional:-->
        <TicketCreateTimeNewerDate>?</TicketCreateTimeNewerDate>
        <!--Optional:-->
        <TicketCreateTimeOlderDate>?</TicketCreateTimeOlderDate>
        <!--Optional:-->
        <TicketChangeTimeOlderMinutes>?</TicketChangeTimeOlderMinutes>
        <!--Optional:-->
        <TicketChangeTimeNewerMinutes>?</TicketChangeTimeNewerMinutes>
        <!--Optional:-->
        <TicketChangeTimeNewerDate>?</TicketChangeTimeNewerDate>
        <!--Optional:-->
        <TicketChangeTimeOlderDate>?</TicketChangeTimeOlderDate>
        <!--Optional:-->
        <TicketCloseTimeOlderMinutes>?</TicketCloseTimeOlderMinutes>
        <!--Optional:-->
        <TicketCloseTimeNewerMinutes>?</TicketCloseTimeNewerMinutes>
        <!--Optional:-->
        <TicketCloseTimeNewerDate>?</TicketCloseTimeNewerDate>
        <!--Optional:-->
        <TicketCloseTimeOlderDate>?</TicketCloseTimeOlderDate>
        <!--Optional:-->
        <TicketPendingTimeOlderMinutes>?</TicketPendingTimeOlderMinutes>
        <!--Optional:-->
        <TicketPendingTimeNewerMinutes>?</TicketPendingTimeNewerMinutes>
        <!--Optional:-->
        <TicketPendingTimeNewerDate>?</TicketPendingTimeNewerDate>
        <!--Optional:-->
        <TicketPendingTimeOlderDate>?</TicketPendingTimeOlderDate>
        <!--Optional:-->
        <TicketEscalationTimeOlderMinutes>?</TicketEscalationTimeOlderMinutes>
        <!--Optional:-->
        <TTicketEscalationTimeNewerMinutes>?</TTicketEscalationTimeNewerMinutes>
        <!--Optional:-->
        <TicketEscalationTimeNewerDate>?</TicketEscalationTimeNewerDate>
        <!--Optional:-->
        <TicketEscalationTimeOlderDate>?</TicketEscalationTimeOlderDate>
        <!--Optional:-->
        <ArchiveFlags>?</ArchiveFlags>
        <!--0回以上繰り返し:-->
        <OrderBy>?</OrderBy>
        <!--0回以上繰り返し:-->
        <SortBy>?</SortBy>
        <!--0回以上繰り返し:-->
        <CustomerUserID>?</CustomerUserID>
     </TicketSearch>
```

11.7.2. サンプル:

11.7.2.1. ウェブサービス構成

下記は基礎的ですが、すべてのチケット・コネクタ・オペレーションを使用するためのYAML フォーマットによる完全なウェブサービス構成ファイルです。OTRSの中でそれを使用するために内容をコピーする必要があります。それをファイルへ保存して GenericTicketConnector.ymlと呼ばれるファイルを保存します。またそれを、概要画面から「ウェブサービスの追加」アクションをクリックすることによって管理者パネルのウェブサービス画面でOTRSへインポートします。そして追加画面で「ウェブサービスのインポート」アクションをクリックします。

```
---
Debugger:
  DebugThreshold: debug
  TestMode: 0
Description: Ticket Connector Sample
FrameworkVersion: 3.2.x CVS
```

```
Provider:
  Operation:
    SessionCreate:
      Description: Creates a Session
      MappingInbound: {}
      MappingOutbound: {}
      Type: Session::SessionCreate
    TicketCreate:
      Description: Creates a Ticket
      MappingInbound: {}
      MappingOutbound: {}
      Type: Ticket::TicketCreate
    TicketUpdate:
      Description: Updates a Ticket
      MappingInbound: {}
      MappingOutbound: {}
      Type: Ticket::TicketUpdate
    TicketGet:
      Description: Retrieve Ticket data
      MappingInbound: {}
      MappingOutbound: {}
      Type: Ticket::TicketGet
    TicketSearch:
      Description: Search for Tickets
      MappingInbound: {}
      MappingOutbound: {}
      Type: Ticket::TicketSearch
  Transport:
    Config:
      MaxLength: 100000000
      NameSpace: http://www.otrs.org/TicketConnector/
    Type: HTTP::SOAP
RemoteSystem: ''
Requester:
  Transport:
    Type: ''
```

11.7.2.2. パールSOAP依頼者

次のコードは、一般インターフェース経由でOTRSに接続することができるパール・スクリプトです。チケット・コネクタによって提供されるオペレーションを行なうために、二つのパールCPANモジュール、SOAP::Lite および Data::Dumper を使用します。sciptを実行しようとする前にこれらのモジュールを使用するのにあなたの環境が利用可能であることを確かめてください。

```
#!/usr/bin/perl -w
# --
# otrs.SOAPRequest.pl - sample to send a SOAP request to OTRS Generic Interface Ticket
 Connector
# Copyright (C) 2001-2015 OTRS AG, http://otrs.com/
# --
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU AFFERO General Public License as published by
# the Free Software Foundation; either version 3 of the License, or
# any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU Affero General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
# or see http://www.gnu.org/licenses/agpl.txt.
# --

use strict;
```

```
use warnings;

# use ../ as lib location
use File::Basename;
use FindBin qw($RealBin);
use lib dirname($RealBin);

use SOAP::Lite;
use Data::Dumper;

# ---
# Variables to be defined.

# this is the URL for the web service
# the format is
# <HTTP_TYPE>:://<OTRS_FQDN>/nph-genericinterface.pl/Webservice/<WEB_SERVICE_NAME>
# or
# <HTTP_TYPE>:://<OTRS_FQDN>/nph-genericinterface.pl/WebserviceID/<WEB_SERVICE_ID>
my $URL = 'http://localhost/otrs/nph-genericinterface.pl/Webservice/GenericTicketConnector';

# this name space should match the specified name space in the SOAP transport for the web
 service.
my $NameSpace = 'http://www.otrs.org/TicketConnector/';

# this is operation to execute, it could be TicketCreate, TicketUpdate, TicketGet,
 TicketSearch
# or SessionCreate. and they must to be defined in the web service.
my $Operation = 'TicketCreate';

# this variable is used to store all the parameters to be included on a request in XML
 format. Each
# operation has a determined set of mandatory and non mandatory parameters to work
 correctly. Please
# check the OTRS Admin Manual in order to get a complete list of parameters.
my $XMLData = '
<UserLogin>some user login</UserLogin>
<Password>some password</Password>
<Ticket>
    <Title>some title</Title>
    <CustomerUser>some customer user login</CustomerUser>
    <Queue>some queue</Queue>
    <State>some state</State>
    <Priority>some priority</Priority>
</Ticket>
<Article>
    <Subject>some subject</Subject>
    <Body>some body</Body>
    <ContentType>text/plain; charset=utf8</ContentType>
</Article>
';

# ---

# create a SOAP::Lite data structure from the provided XML data structure.
my $SOAPData = SOAP::Data
    ->type( 'xml' => $XMLData );

my $SOAPObject = SOAP::Lite
    ->uri($NameSpace)
    ->proxy($URL)
    ->$Operation($SOAPData);

# check for a fault in the soap code.
if ( $SOAPObject->fault ) {
    print $SOAPObject->faultcode, " ", $SOAPObject->faultstring, "\n";
}

# otherwise print the results.
else {

    # get the XML response part from the SOAP message.
```

```
    my $XMLResponse = $SOAPObject->context()->transport()->proxy()->http_response()-
>content();

    # deserialize response (convert it into a perl structure).
    my $Deserialized = eval {
        SOAP::Deserializer->deserialize($XMLResponse);
    };

    # remove all the headers and other not needed parts of the SOAP message.
    my $Body = $Deserialized->body();

    # just output relevant data and no the operation name key (like TicketCreateResponse).
    for my $ResponseKey ( keys %{$Body} ) {
        print Dumper( $Body->{$ResponseKey} );
    }
}
```

12. OTRSスケジューラ

OTRSスケジューラは背景のタスクを実行する独立したシステムプロセスです。これらのプロセスはUnix / Linuxシステムではデーモンとして、またWindows環境上ではサービスとして知られています。それは独立していますが、スケジュールが全て単独で独立しているという意味ではありません。それは十分にOTRSに統合されていて、必要な時に各タスクを完了するためにいかなるOTRSモジュールをも使うことができます。

正常な理由のためにスケジューラ・プロセスはときどきリスタートする必要があります。これは、一日に一度スケジューラ・プロセス自体によって自動的に行われます。しかし、必要な時、SysConfigを使用して「Scheduler::RestartAfterSeconds」設定の編集すればそれを調節することができます。

OTRSスケジューラは完全に自動プロセスです。唯一の人間の介在が必要とされることは、そのステータスを周期的にチェックし、必要な時にそれをスタートするか停止することです。

注記

スケジューラが何らかの理由で停止されたら、スケジューラが停止した時に登録された全ての保留タスクと新規タスクは、スケジューラが再開するとすぐに実行されます。(タスクがその後実行される準備ができていない限り)

12.1. スケジューラ・グラフィカル・インターフェース

スケジューラは、それが停止しない限り、OTRSグラフィカルユーザーインターフェイスにおいて不可視です。

12.1.1. スケジューラ停止中の通知

スケジューラが停止中をシステムが検知する場合、2つの異なるタイプの通知があります。この検知はスケジューラ・プロセスの更新頻度に基づきます。現在時刻と最終プロセス更新時刻の時間差が 2 倍以上ならば、OTRS通知エリア内にプロセス更新頻度警告メッセージが表示されます。それが4倍以上のプロセス頻度ならば、警告に変わって警戒メッセージが表示されます。

スケジューラ・プロセス更新時間は、「Scheduler::PIDUpdateTime」設定中のSysConfigによって構成することができます。警告メッセージを見ても必ずしも処置を講ずる必要はありません。

しかし、スケジューラ・プロセスが走っているかどうかをチェックすることは強く推奨します。警戒メッセージを見たら、スケジューラが停止している可能性が高いのでスタートすべきです。

データベースに有効なウェブサービスが登録されていたり、または「管理者」グループのユーザだけに表示されている場合、デフォルトでスケジューラは通知を有効ではありません。

通知を不能にするためには (推奨しませんが) あるいは、通知グループを変更・追加するためには、SysConfigの中の設定「Frontend::NotifyModule###800-Scheduler-Check」を編集してください。

図:スケジューラ通知

12.1.2. スタート・スケジューラ

スケジューラ停止中通知リンク(警告か警戒のいずれか)をクリックすることによって、再びスケジューラ・プロセスをを再開させるためにダイアログ・ボックスを開きます。ダイアログ中のチェックボックスを適切にクリックすることでスケジューラは正常にスタートするか強制的にスタートさせられます。

注記

強制的なスケジューラ・スタートは、前のスケジューラ・プロセスが異常に終了し、プロセスIDがデータベース中でまだ登録されている場合のみ、必要です。

スケジューラ・プロセスを十分にコントロールをするために、以下の述べるコマンドラインツールをつかって、その実際の状態をチェックして下さい。

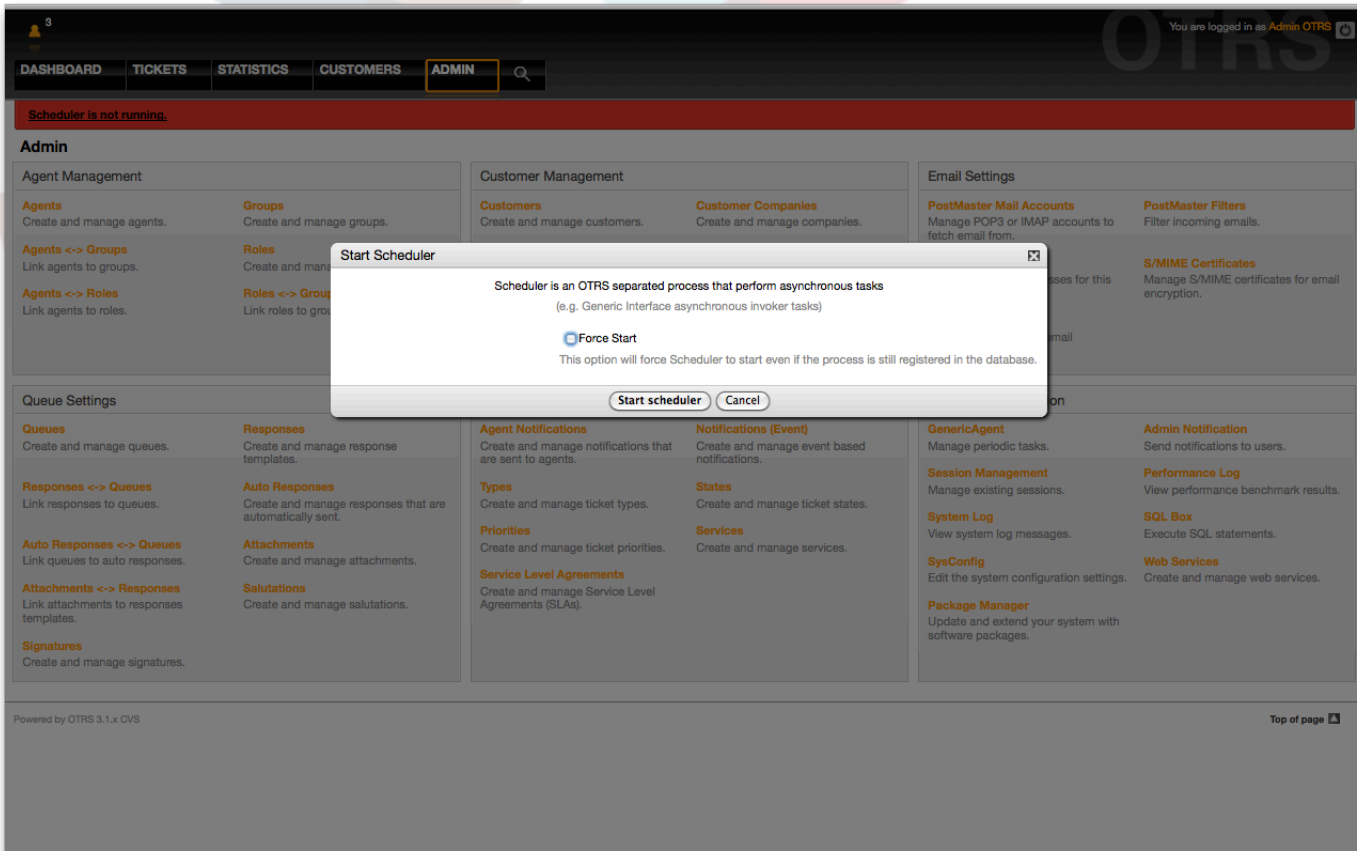


図:スケジューラの開始

12.2. スケジューラ・コマンドライン・インタフェース

スケジューラ・コマンドライン・ツールによってスケジューラ・プロセス(スタート・ストップ)をコントロールし、スケジュールの状態を尋ねます。さらにオペレーティング・システムによってコントロールされるプロセスを登録するツールがあります。

2セットのCLIのツール、すなわちひとつはUnix/Linux用でもう一方はMicrosoft Windows用です。

12.2.1. Unix/Linux

12.2.1.1. Scheduler init.d files

init.d ファイルは、スタートアップとシャット・ダウンでオペレーティング・システムによって呼び出される特別なスクリプトです。

OTRSは、オペレーティング・システムによってOTRSスケジューラ・プロセスを自動的に始める/止めるためにinit.d スクリプトを提供します。これらのスクリプトはOTRS_HOME/scriptsの下にあります。

init.d スクリプトは、ご利用のオペレーティング・システムに応じて正しい位置にコピーする必要があります。それらは適切な許可を持つ必要があります。また、いくつかの内部変数を適切に動くようにセットする必要があります。システムをセット・アップにOTRS RPMを使用したならば、これは自動的に処理されます。

init.d スクリプト内部変数

- **OTRS_HOME** - OTRSインストールのパス。
- **User** - the apache プロセスユーザ名。
- **Group** - the apache OTRSインストールのグループ名。

> **注記**
>
> 現在、OTRSは、Linuxプラットフォームのinit.d スクリプトだけを提供します。

表4.7 テーブル4.7. init.d スクリプトのリストとサポートされるオペレーティング・システム

Init Script	サポートするOS
otrs-scheduler-linux	Red Hat, Fedora, CentOS, SUSE, openSUSE, Debian, Ubuntu
otrs-scheduler-gentoo-init.d, otrs-scheduler-gentoo-conf.d	Gentoo

例4.29 Linux上でOTRSを開始するサンプル

```
shell> /etc/init.d/otrs-scheduler-linux start
```

可能なアクション

- **start** - OTRSスケジューラ・プロセスを開始
- **stop** - OTRSスケジューラ・プロセスを停止。
- **restart** - OTRSスケジューラ・プロセスを再開。

- **status** - OTRSスケジューラ・プロセス・ステータスをクエリー

スケジューラは、そのプロセスIDを登録することができるようにデータベースを必要とします。というのは、この理由は次のものに必要であるからです:

- データベース・プロセスの活動開始以後、スケジューラ・プロセスを開始するためにスケジューラinit.d スクリプトを実行します。
- データベース・プロセスがシャット・ダウンする前に、スケジューラを停止ためにスケジューラinit.d スクリプトを実行します。

注記

スケジューラがシステム起動時に開始することを望むなら、オペレーティング・システムのドキュメンテーションを読んで、init.dスクリプトを置く正しい場所、それらの自動的に開始するための構成方法及び起動順の設定方法を見つけて下さい。

12.2.1.2. スケジューラ・デーモン・ファイル

これは、実行すべきタスクをチェックしながらバックグラウンドで稼働し続けるスケジューラの部分です。さらに、それは、プロセスをコントロールするために主機能を提供します。

全てのUnix/Linuxはファイル **OTRS_HOME/bin/otrs.Scheduler.pl**を使用します。

例4.30 OTRSスケジューラを開始するサンプル

```
shell> OTRS_HOME/bin/otrs.Scheduler.pl -a start
```

利用可能なオプション

- **-a** アクション

  #### 可能な値

  - **start** - スケジューラ・プロセスを開始。
  - **stop** - スケジューラ・プロセスを停止。
  - **status** - スケジューラ・プロセス状態のクエリー。
- **-f** スケジューラ・プロセスの強制的開始または停止。

例4.31 OTRSスケジューラを強要停止するサンプル

```
shell> OTRS_HOME/bin/otrs.Scheduler.pl -a stop -f 1
```

注記

スケジューラが走っておらず、プロセスがまだ登録されている時に、スケジューラを強制終了させることは、データベースからプロセスIDを取り除くことになります。

スケジューラが走っておらず、プロセスがまだ登録されていれば、スケジューラを強制終了させることは、スケジューラプロセスを開始することになります。

強制開始または強制停止は、プロセス開始がプロセス更新時間が失効する前になされる必要な場合のみ必要です。そうでなければ
データベース内の失効エントリーは通常開始で破棄されます。

12.2.2. Windows

12.2.2.1. スケジューラ・サービス・インストーラー

Windowにおいては、OTRSスケジューラがWindow・サービス・コントロール・マネージャー(SCM)によってサービスとして登録されます。OTRSスケジューラを持つためには、SCMによってコントロールされるプロセスがこのサービスを登録する必要があります。Windowインストーラを使用した場合インストーラによって処理され、手動のアクションは必要ありません。

OTRSは、SCMへOTRSスケジューラを登録するか登録解除するためにスクリプト**OTRS_HOME/bin/otrs.Scheduler4WinInstaller.pl**を提供します。

注記

登録および登録解除することはウィンドウズ上の管理者グループのメンバーによってのみ行なわれます。サービスを正確に登録する前に、cmd.exeを右クリックし、「管理者として運用」を選択するします。

例4.32 Windowsサービスとして**OTRS**スケジューラを登録するサンプル

```
shell> perl OTRS_HOME/bin/otrs.Scheduler4WinInstaller.pl -a
install
```

利用可能なオプション

- **-a** アクション

可能な値

- **install**- ウィンドウズSCMへスケジューラ・プロセスをインストールしてください。
- **remove**- ウィンドウズSCMからスケジューラ・プロセスを取り除いてください。

WindowsSCMへインストール後、OTRSスケジューラ・プロセスはウィンドウズ上で他のサービスのように使用することができます。それは開始、停止、再開することができ、手動あるいは自動での開始を構成できます。

注記

ウィンドウズ・サービスおよびウィンドウズSCMに関してもっと学習するために、ウィンドウズ・ドキュメンテーションおよびマイクロソフト・オンライン・ヘルプを読んでください。

12.2.2.2. スケジューラ・サービス・ファイル

これは、実行すべきタスクをチェックしながらバックグラウンドで稼働し続けるスケジューラの部分です。さらに、それは、プロセスをコントロールするために主機能を提供します。

Microsoft Windows uses the file **OTRS_HOME/bin/otrs.Scheduler4Win.pl**.

例4.33 ウィンドウズ上の**OTRS**スケジューラを開始するサンプル

```
shell> OTRS_HOME/bin/otrs.Scheduler4Win.pl -a start
```

利用可能なオプション

- **-a** アクション

可能な値

- **start** - スケジューラ・プロセスを開始。
- **stop** - スケジューラ・プロセスを停止。
- **status** - スケジューラ・プロセス状態のクエリー。

- **-f** スケジューラ・プロセスの強制的開始または停止。

例**4.34 Windows**上で**OTRS**スケジューラを強制停止するサンプル

```
shell> OTRS_HOME/bin/otrs.Scheduler4Win.pl -a stop -f 1
```

注記

スケジューラが走っておらず、プロセスがまだ登録されている時に、スケジューラを強制終了させることは、データベースからプロセスIDを取り除くことになります。

スケジューラが走っておらず、プロセスがまだ登録されていれば、スケジューラを強制開始することは、スケジューラプロセスを開始することになります。

強制開始または強制停止は、プロセス開始がプロセス更新時間が失効する前になされる必要な場合のみ必要です。そうでなければデータベース内の失効エントリーは通常開始で破棄されます。

第5章 カスタマイズ

1. アクセス・コントロール・リスト(ACL)

1.1. 導入

OTRS 2.0から、アクセス・コントロール・リスト(ACL)は、チケット、モジュール、キューなどへの制御アクセスに使用することができます。また、ある状況でチケット(完了する、移動するなど)に対するアクションに影響を及ぼすことができます。ACLは役割とグループの既存の許可システムを補足するために使用することができます。ACLを使用すると、システム内の基本的なワークフローは、チケット属性に基づいてマップすることができます。

まだ、ACLはSysConfigインターフェースを使用して作成することができません。直接Kernel/Config.pmファイルにそれらを入力しなければなりません。本章にはいくつかのACL例がありますので、それらはあなたをACLの諸定義を定義する過程へ導き、あらゆる可能で重要なACL設定の参考となるでしょう。

1.2. サンプル

例5.1 ACLはチケット優先度5のチケットのみをキューへの移動を許可します。

この例は、あなたにACLの基本構造を示します。最初に、それは名前を持つ必要があります。ここでは「ACL-Name-2」とします。まず、ACLは実行の前に数値的にソートされることにご留意下さい。それゆえACLの名前を注意深く使用して下さい。

次に、チケット用のフィルタである「プロパティ」セクションがあります。ここで定義された基準はすべて、ACLが適用されるか否かを判断するチケットに適用されます。例において、チケットが「未対応の」キューかあるいは優先度が「5　非常に高い」にあれば、チケットは一致するでしょう。これはフォーム内での変更により影響を受けます。(「未対応の」キューかあるいは優先度が「3：普通」のチケットが、その後、優先度のドロップダウンが選択され、優先度が「5：非常に高い」へ変更されたような場合)

最後に、「可能」セクションは画面への修正を定義します。この場合、利用可能キューから、「警告」キューだけがチケット・画面の中で選択することができます。

```
# ticket acl
$Self->{TicketAcl}->{'100-Example-ACL'} = {
    # マッチプロパティ
    Properties => {
        # 現在のチケット・マッチ・プロパティ
        Ticket => {
            Queue => ['Raw'],
            Priority => ['5 very high'],
        }
    },
    # 可能なオプション(ホワイトリスト)を返します
    Possible => {
        # 可能なオプション(ホワイトリスト)を返します
        Ticket => {
            Queue => ['Alert'],
        },
    },
};
```

例5.2 データベースに格納された優先度5のチケットだけのキューへ移動を許可するACL

この例は先のものに非常に似ています。しかしながら、この例では、「生で」かつ優先度が「5：非常に高い」チケットのみが両者ともにデータベースに保存されて、マッチします。チケットがデータベースの中で実際に更新される前に、この種のACLはフォームの変更を考慮しません。

```
# ticket acl
$Self->{TicketAcl}->{'100-Example-ACL'} = {
    # マッチプライオリティ
    PropertiesDatabase => {
        # 現在のチケット・マッチ・プロパティ
        Ticket => {
            Queue => ['Raw'],
            Priority => ['5 very high'],
        }
    },
    # 可能なオプション（ホワイトリスト）を返します
    Possible => {
        # 可能なオプション（ホワイトリスト）を返します
        Ticket => {
            Queue => ['Alert'],
        },
    },
};
```

例5.3 未対応のキュー中のチケットの完了を不能にし、閉じるボタンを隠すACL

この例では、どのようにチケット・フィールド（の状態）が、（フォームを選択する 1 つ以上の可能な値で）フィルターされるかを理解できるでしょう。あるチケット用に実行することができるアクションを制限することも可能です。この場合、チケットは完了できません。

```
$Self->{TicketAcl}->{'101-Second-Example-ACL'} = {
    # マッチプロパティ
    Properties => {
        # 現在のチケット・マッチ・プロパティ
        Ticket => {
            Queue => ['Raw'],
        }
    },
    # 可能なオプション（ホワイトリスト）を返します
    Possible => {
        # 可能なオプション（ホワイトリスト）を返します
        Ticket => {
            State => ['new', 'open', 'pending reminder'],
        },
        # 可能なアクション・オプション
        Action => {
            AgentTicketBounce        => 1,
            AgentTicketClose         => 0,
            AgentTicketCompose       => 1,
            AgentTicketCustomer      => 1,
            AgentTicketForward       => 1,
            AgentTicketFreeText      => 1,
            AgentTicketHistory       => 1,
            AgentTicketLink          => 1,
            AgentTicketLock          => 1,
            AgentTicketMerge         => 1,
            AgentTicketMove          => 1,
            AgentTicketNote          => 1,
            AgentTicketOwner         => 1,
            AgentTicketPending       => 1,
            AgentTicketPhone         => 1, # 分かれているアクションを非表示にするためだけに使用されます
            AgentTicketPhoneInbound  => 1,
            AgentTicketPhoneOutbound => 1,
            AgentTicketPrint         => 1,
            AgentTicketPriority      => 1,
            AgentTicketResponsible   => 1,
            AgentTicketWatcher       => 1,
            AgentTicketZoom          => 1,
            AgentLinkObject          => 1, # リンク・アクションを非表示にするためだけに使用されます
        },
    },
};
```

例5.4 状態：「閉鎖成功」を常に削除するACL

ここの例では、否定フィルタ（状態：「完了（成功）」を取り除く）をどのようにして定義可能か例示します。チケット用のマッチ・プロパティを定義しなければ、どんなチケットでもマッチすることもわかります。例えばACLが常に当てはまります。これは、もしデフォルトで特定の値を非表示にしたくかつ、特殊な環境（例えば担当者が特定のグループにいるならば）だけにそれらを可能にしたいなら活用可能になります。

```
$Self->{TicketAcl}->{'102-Third-ACL-Example'} = {
    # マッチ・プロパティ
    Properties => {
        # 現在のチケットのマッチ・プロパティ(常にマッチ)
    },
    # 可能なオプションに戻る
    PossibleNot => {
        # 可能なチケット・オプション
        Ticket => {
            State => ['closed successful'],
        },
    },
};
```

例5.5 "HW"で始まるキューが作成されるハードウェア・サービス・チケットのみを表示するACL

ここでは、マッチするチケットと利用可能なオプションのフィルタリングのための正規表現を使用することができるかについても例示します。

```
$Self->{TicketAcl}->{'Only-Hardware-Services-for-HW-Queues'} = {
    # マッチ・プロパティ
    # note we don't have "Ticket => {" because there's no ticket yet
    Properties => {
    Queue => {
        Name => ['[RegExp]HW'],
        }
    },
    # 可能なオプションに戻る
    Possible => {
        # 可能なチケット・オプション
        Ticket => {
            Service => ['[RegExp]^(Hardware)'],
        },
    },
};
```

1.3. 参照

下記の例において、ACLで使用可能な全パラメーターのリストがあります。

ACLをプロセス・チケットに使用する方法の詳細な記述については、ProcessManagement文書でACLについてのセクション を参照してください。

例5.6 全ての可能かつ重要なACL設定を示すリファレンス（参照）

```
# ticket acl
$Self->{TicketAcl}->{‘200-ACL-Reference’} = {
    # マッチ・プロパティ (フォームからの現在の値)
    Properties => {

        # 使用されるフロントエンド・モジュール
        Frontend => {
            Action => [‘AgentTicketPhone’, ‘AgentTicketEmail’],
        },

        # ログインした担当者
```

```
        User => {
            UserLogin => [‘some login’],
            Group_rw => [
                ‘hotline’,
            ],
            Role => [
                ‘admin’,
            ],
            # …
        },

        # ログインした顧客
        CustomerUser => {
            UserLogin => [‘some login’],
            Group_rw => [
                ‘hotline’,
            ],
            Role => [
                ‘admin’,
            ],
            # …
        },

        # プロセス・プロパティ
        Process => {
            ProcessEntityID        => [‘P1’],   # 現在のチケットがその一部であるプロセス
            ActivityEntityID       => [‘A1’],   # 現在のチケットのアクティビティ
            ActivityDialogEntityID => [‘AD1’],  # 担当者/顧客が使用している現在のActivityDialog
        },

        # チケット・プロパティ
        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 => {
            # 名前はDynamicField_<field_name> formatになければなりません。
            # [ … ] の値は常に未翻訳の内部データキーでなければなりません。
            #   動的フィールド定義で指定され、かつ
            #   ユーザに非表示のデータ値です。
            DynamicField_Field1          => ['some value'],
            DynamicField_OtherField      => ['some value'],
            DynamicField_TicketFreeText2 => ['some value'],
            # …
        },
        # 別の方法としてはチケット・プロパティはチケット・ハッシュで指定もできます。
        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'],
            # …
        },
    },

    # マッチプロパティ (データベースからの既存値)
    PropertiesDatabase => {
        # プロパティのセクションを御覧ください。ここと同じconfigが使用されています。
        # …
    }

    # 可能なオプションに戻る (ホワイトリスト)
    Possible => {
        # 可能なチケットオプション (ホワイトリスト)
        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'],
            # …
        },

        # 担当者/顧客が 1 つのプロセス・チケットで利用可能な
        #   ActivityDialogs数を制限します。
        ActivityDialog => ['AD1', 'AD3'],

        # 開始可能なプロセス数を制限します。
        Process => ['P1', 'P2'],

        # 可能なアクション・オプション (ホワイトリスト)
        Action => {
            AgentTicketBounce        => 1,
```

```
            AgentTicketClose         => 1,
            AgentTicketCompose       => 0,
            AgentTicketCustomer      => 0,
            AgentTicketForward       => 0,
            AgentTicketFreeText      => 1,
            AgentTicketHistory       => 1,
            AgentTicketLink          => 0,
            AgentTicketLock          => 1,
            AgentTicketMerge         => 0,
            AgentTicketMove          => 1,
            AgentTicketNote          => 1,
            AgentTicketOwner         => 1,
            AgentTicketPending       => 1,
            AgentTicketPhone         => 1, # 分割アクションを非表示にするためだけに使用します。
            AgentTicketPhoneInbound  => 0,
            AgentTicketPhoneOutbound => 1,
            AgentTicketPrint         => 1,
            AgentTicketPriority      => 0,
            AgentTicketResponsible   => 1,
            AgentTicketWatcher       => 1,
            AgentTicketZoom          => 1,
            AgentLinkObject          => 1, # リンク・アクションを非表示にするためだけに使用します。
        },
    },
    # オプション（ブラックリスト）を削除します
    PossibleNot => {
        # See section "Possible"
        # …
    },
};
```

注記

ACLとマッチしている間に、CustomerUserIDパラメーターが送られると、ACLメカニズムは、CustomerUserハッシュを満たすCustomerUser詳細を集めるために、供給されたCustomerUserIDを使用して、定義されたACLを比較します。またそれは、プロパティ（特性）・マッチ用のチケット・ハッシュ中の顧客情報を無効に(override)します。この計算もPropertiesDatabase部分のためになされるが、その一方で、データを集めるために基礎としてチケット・顧客(the Ticket Customer)を使用しています。

顧客・インターフェースでは、CustomerUserIDが現在の記録された顧客ユーザとともに常に送信されることに注目してください。

チケット検索画面(AgentTicketSearchとCustomerTicketSearch)では、ACLによって影響を受ける属性のみが動的フィールドであることにご留意下さい。チケット検索画面(AgentTicketSearchとCustomerTicketSearch)では、ACLによる影響を受けた数少ない属性が動的フィールドであることにご留意下さい。すなわち、この(検索)画面は、あなたがその他の属性を制限できないことを意味します。（例えばチケットタイプ、状態、キュー等）

2. 工程管理

2.1. 導入

OTRSのこの機能により、チケット・システムでプロセス(ワークフロー)をモデル化することが可能です。この基本概念は再現するプロセスを定義可能にし、異なる人々に作業項目を任せるためのものです。またある基準に基づく異なる指示のプロセス進行をリードします。

2.2. サンプル・プロセス

より実地的なサンプルで見てみましょう。ここでは本の注文プロセスを定義します:

2.2.1. 要求の記録

受注前に、従業員によって書物の需要が記録されます。以下の本がここでの例に必要となります:

```
タイトル：Prozessmanagement für Dummies、著者：Thilo Knuppertz、ISBN：3527703713
```

2.2.2. マネージャーによる承認

従業員の部長は注文に関して以下を決定する必要があります。否認の場合には、理由がマネージャーによって記録されるべきです。承認の場合には、オーダーが購買部に渡されます。

2.2.3. 購買部による処理

購入するとは、本を最良の条件で本を注文する場所を見つける職務です。品切れの場合、注文に記録します。購入が成約した場合はサプライヤー、購入価格、配達日を記録します。

2.2.4. 郵便室による処理

貨物が会社に入荷すると、貨物の入荷受付部は、その貨物をチェックし受領日を記録します。従業員はこの注文が到着を知らされ、貨物を取りに行く準備をするでしょう。

2.3. 例の実行

チケットが、このワークフロー中で、
変更ノートを受け取ることができる添付書類のように作用すると考えれば、既にプロセス・チケットの明瞭な青写真を持っていることになります。

サンプル・プロセスの分析から、次の必須アイテムを識別することができます。

- データを記録する可能性、それをアクティビティ・ダイアログと呼びます。
- データ変更されたことに対して自動的に反応するチェック機構、それを推移 (Transitions)と呼びます。
- プロセス・チケットの推移が成功した後でプロセス・チケットに適用可能な変更、それを推移アクション (Transition Actions) と呼びます。

さらにそれほど明白でないかもしれない追加項目を必要とします:

- 利用可能なアクティビティ・ダイアログをひとつ以上提供する可能性。ここでの例においては、マネージャーが「承認」「拒絶」の選択を持っていなければならない場合にこれ (提供する可能性) が必要です。これをアクティビティと呼びます。

さて、アクティビティ、アクティビティ・ダイアログ、推移と推移アクションにより、私たちは、私たちの例の個々のステップを形成するための必要なツールを持つことになります。私たちは、各々のワークフォローにとってステップの順番が指定されうるエリアをまだ見つけていません。これをプロセスと呼びましょう。これらのすべての実体 (エンティティ) を後で参照することができるようにするために、それらを括弧中の略語で指定します。この省略は EntityIDs と呼ばれる内部識別メカニズムに基づきます。

エンティティIDs(EntityIDs)は、1個あるいは2個の文字 (プロセス部分あるいは実体に依存します) と連続した番号で組み合わせて作られます。

- プロセス (Process) : 'P1', 'P2' ... 'Pn'.
- アクティビティ (Activity) : 'A1', 'A2' ... 'An'.
- アクティビティ・ダイアログ (Activity Dialog) : 'AD1', 'AD2' ... 'ADn'.
- 推移 (Transition) : 'T1', 'T2' ... 'Tn'.

- 推移アクション (Transition Action) : 'TA1', 'TA2' ... 'TAn'.

プロセスとその部分の生成がシステムを準備するために必要とする前に、いくつかのSysConfigオプションのセット、キュー、ユーザー、動的フィールドを定義する必要があります。

次のキューを作成します：

- 管理： Management
- 顧客： Employees
- 購買：Purchasing
- 郵便局： Post office

次のユーザーを作成します。

- 管理者：Manager
- 顧客：Employee

次の動的フィールドを作成します。

- タイトル

ラベル	タイトル
タイプ	テキスト
オブジェクト	チケット

- 著者

ラベル	著者
タイプ	テキスト
オブジェクト	チケット

- ISBN

ラベル	ISBN
タイプ	テキスト
オブジェクト	チケット

- 状態

ラベル	状態
タイプ	ドロップダウン
オブジェクト	チケット
可能な値	• 承認 • 承認拒否 • 承認済 • 注文拒否 • 発注済 • 貨物受取済

ノート: フィールド設定の「キー」と「値」は的確に使用して下さい。

- サプライヤー

ラベル	サプライヤー
タイプ	テキスト
オブジェクト	チケット

- 価格

ラベル	価格
タイプ	テキスト
オブジェクト	チケット

- 配達日 (DeliveryDate)

ラベル	配達日
タイプ	日付
オブジェクト	チケット

- 受領日 (DateOfReceipt)

ラベル	受領日
タイプ	日付
オブジェクト	チケット

次のSysConfigの設定をセットします：

- 'Ticket::Responsible': はい
- 'Ticket::Frontend::AgentTicketZoom###ProcessWidgetDynamicFieldGroups':

キー:	コンテンツ:
本	タイトル、著者、ISBN
一般	状態
注文	価格、サプライヤー、配達日
配送	受領日 (DateOfReceipt)

- 'Ticket::Frontend::AgentTicketZoom###ProcessWidgetDynamicField':

キー:	コンテンツ:
著者	1
受領日 (DateOfReceipt)	1
配達日 (DeliveryDate)	1
ISBN	1
価格	1
状態	1
サプライヤー	1
タイトル	1

それでは、実際のプロセス管理職員で開始しましょう。次のステップで、必要とする個々の実体を定義します。

2.3.1. プロセス(コンテナーとしての)

管理者パネル中のシステム管理ボックスの中の「プロセス・マネジメント」リンクをクリックする際には、新しいプロセスを作成することが必要です。これはプロセス・マネジメント概要への通じます。プロセスの生成の後、他のすべての実体(あるいはプロセス部分)を作成することができます。

注記

1つのプロセスの中で定義されたアクティビティ、アクティビティ・ダイアログ、推移および推移アクションは、システムでのすべてのプロセスに利用可能になります。

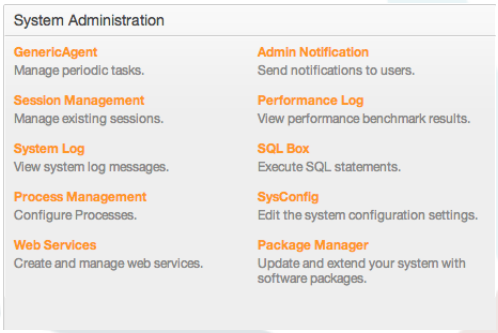


図:OTRS管理者画面 - システム管理

アクション・ボックスから「新しいプロセスを作成する」アクションをクリックしてください。

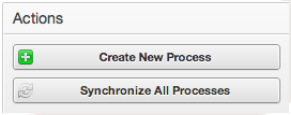


図:新しいプロセスの追加

プロセス情報を記入し、プロセス名と摘要を設定し、全てのタスクが終了するまでプロセスの状態を「不活性」のままにしておきます。プロセスを保存します。

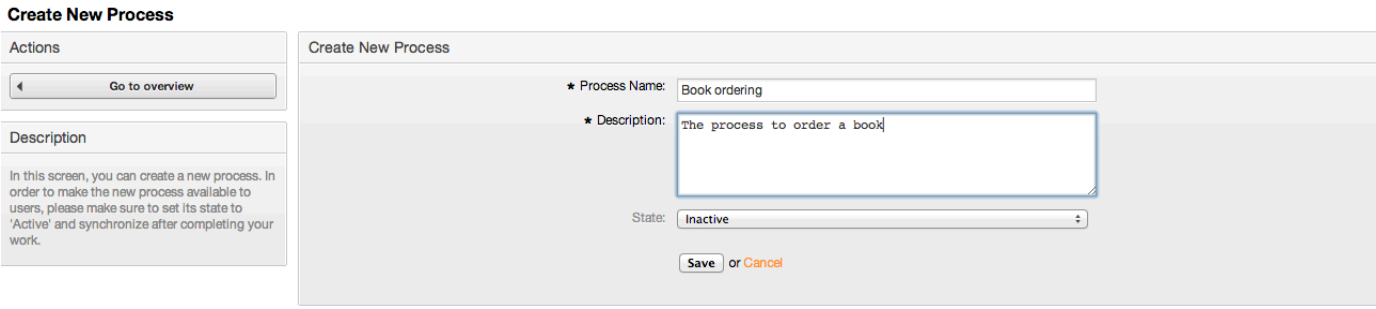


図:新しいプロセスの追加

2.3.2. アクティビティ・ダイアログ

プロセス・マネジメント概要画面にあるの新しいプロセス名をクリックします。それから「利用可能なプロセス構成要素」の中で「アクティビティ・ダイアログ」をクリックします。(このアクションはアクティビティ・ダイアログ・オプションを拡張し、他の全てを壊し押しつぶすでしょう。)次に「新しいアクティビティ・ダイアログを作成」をクリックします。

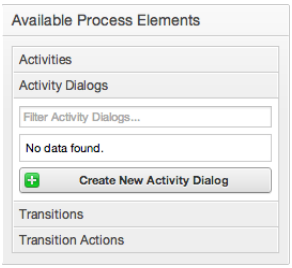


図:新しいアクティビティ・ダイアログ・ボタンの作成

「摘要（短）」フィールドと同様に、開いたポップ・アップ・画面に「アクティビティ・ダイアログ名」を 記入します。この例では全ての他のフィールドをデフォルトのままにしておきます。アクティビティ・ダイアログにフィールドを指定するために、必要なフィールドを「利用可能なフィールド」プールから「指定されたフィールド」へドラッグします。「指定されたフィールド」プールの順序は
　フィールドが画面にあるような順序です。プール内のフィールドをドラッグ＆ドロップして順序を変更し、正しい位置に再配置します。

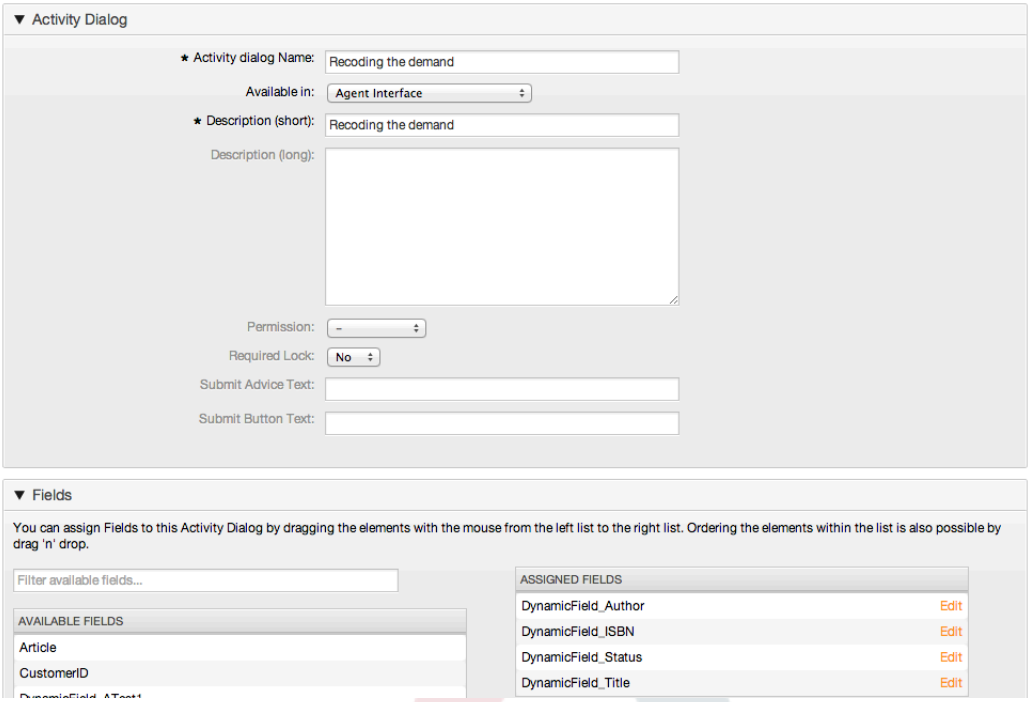


図:新規アクティビティ・ダイアログの追加

フィールドが「指定されたフィールド」プールにドロップされるとすぐに、別のポップ・アップ・画面が開き、フィールドに関するいくつかの詳細が表示されます。デフォルト・オプションはそのままとして、また記事フィールドのみに関しては、記事タイプ・フィールドが「ノート内部：（note-internal）」にセットされることを確かめねばなりません。

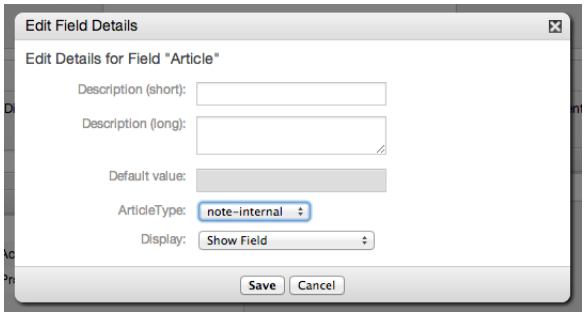


図:フィールド詳細(記事)の編集

全てのフィールドが指定された後で、変更を保存するためにメインのポップアップ画面にて、送信ボタンをクリックします。

この例では、私たちは記事フィールドをコメントに使用します。しかし、別のオプションはTextAreaタイプ動的フィールドを作成できます。下記のラインの言及されたフィールドの残りは、私たちが事前に定義する動的フィールドです。

この画面では動的フィールドは、「DynamicField_Title」の中に接頭辞「DynamicField_」を持つことにご留意願います。タイトルフィールドとチケット・タイトルとを混同しないようにして下さい。

次のアクティビティ・ダイアログを作成してください:

- 「依頼の記録」(AD1)

  注文のために収集されるデータ用の全ての必須フィールドを含むアクティビティ・ダイアログ(タイトル、著者、ISBN)と、「承認」を選択する可能性を持つ状態フィールド。

- 「承認拒否」(AD2)

  コメント・フィールド(記事)を持つアクティビティー・フィールドとオプション「承認拒否」を備えた状態フィールド

- 「承認済」(AD3)

  ここでは「承認済」オプションを持つ状態フィールドが必要となります。

- 「注文拒否」(AD4)

  購入時不可能な注文(在庫切れの本)を拒否することを可能可能にするアクティビティ・ダイアログ。ここでは「注文拒否」オプションを持つ状態フィールドが必要となります。

- 「発注」(AD5)

  購入時のフィールド・サプライヤー、価格、配達日、及び「注文拒否」オプションを持つ状態フィールドを、持つアクティビティ・ダイアログ。

- 「貨物受取」(AD6)

  「受取日のフィールドを備えた郵便室」および「貨物受取」オプションを備えた状態・フィールド用のアクティビティ。

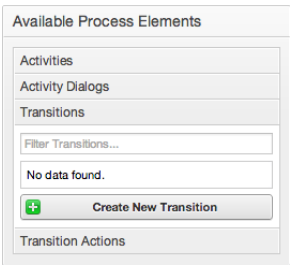
各々のアクティビティ・ダイアログ用の状態フィールドを制限するために、いくつかのACLの追加を必要とします。ACLを追加する場所はKernel/Config.pmの中またはKernel/Config/Filesの中にある新しいperlファイルです。

```
$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. 推移

「推移」で「利用可能なプロセス要素」をクリックして、その後、「新しい推移の作成」をクリックしてください。



図：新規推移ボタンの作成

開いたポップ・アップ・画面中に「推移名」を記入します。それからこの条件では、この例に対してひとつの条件とひとつのフィールドを使います。その両方に対してリンクのタイプを「and」のままにします。そしてフィールド一致タイプ値として「String」を使います。

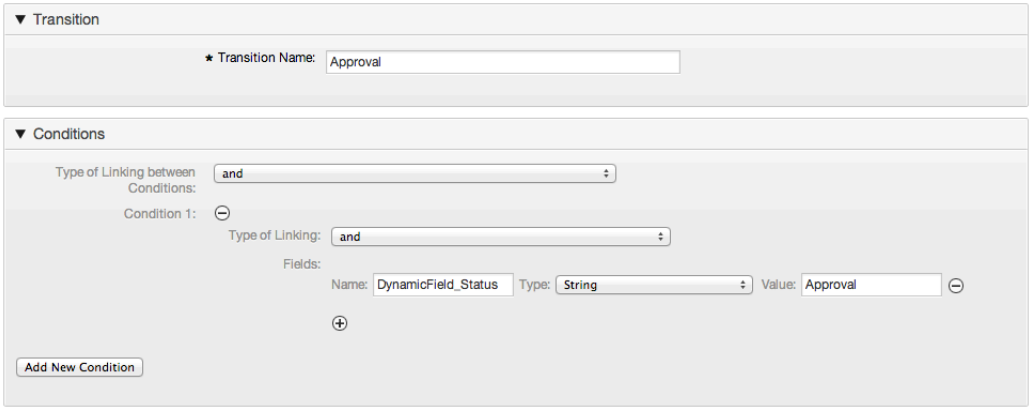


図: 新規推移の追加

全ての条件を設定後、変更を保存するために送信ボタンをクリックします。

次の推移を作成してください:

- 「承認」(T1)

  状態フィールドが「承認」にセットされるかどうかをチェックする推移。

- 「承認拒否」(T2)

  状態フィールドが「「承認拒否」にセットされるかどうかをチェックする推移。

- 「承認済」(T3)

  状態フィールドが着手されるかどうかをチェックする推移は「承認済」

- 「注文拒否」(T4)

  状態フィールドが「注文拒否」にセットされるかどうかをチェックする推移。

- 「発注」(T5)

  状態フィールドが「発注」にセットされるかどうかをチェックする推移。

- 「貨物受取」(T6)

  状態フィールドが「貨物受取」にセットされるかどうかをチェックする推移。

2.3.4. 推移アクション

「利用可能なプロセス要素」中の「推移アクション」をクリックして、次に、「新規推移アクションの作成」をクリックしてください。

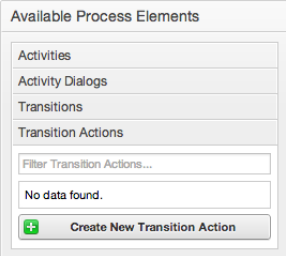


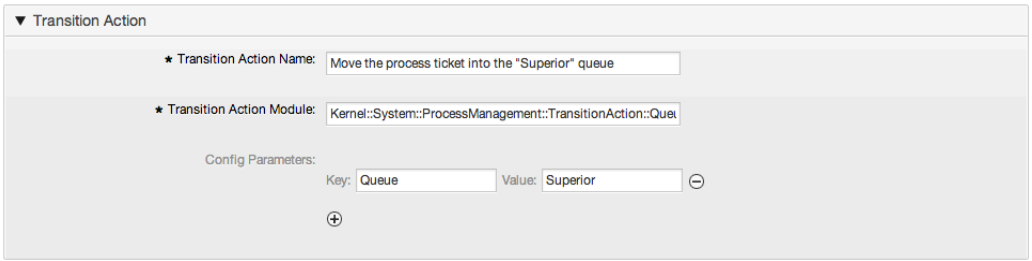
図:新規推移アクションボタンの作成

開かれたポップ・アップ・画面の中で「推移名」および「推移アクション・モジュール」を記入して下さい。それから、必須及び任意のパラメタと値を追加して下さい。

全ての推移アクション・モジュールは、Kernel/System/ProcessManagement/TransitionActionに存在します。また、推移アクションと以下は下記は、バンドルされた推移アクションのリストで、このリリースに含まれます。

- DynamicFieldSet (動的フィールドのセット)
- TicketArticleCreate (チケット記事の作成)
- TicketCustomerSet (チケット・顧客・セット)
- TicketLockSet (チケット・ロック・セット)
- TicketOwnerSet (チケット・所有者・セット)
- TicketQueueSet (チケット・キュー・セット)
- TicketResponsibleSet (チケット・責任者・セット)
- TicketServiceSet (チケット・サービス・セット)
- TicketSLASet (チケット・SLA・セット)
- TicketStateSet (チケット・状態・セット)
- TicketTitleSet (チケット・タイトル・セット)
- TicketTypeSet (チケット・タイプ・セット)

モジュールはそれぞれそれ自身のものおよび異なるパラメーターを持っています。全ての必須及び任意のパラメタを知るためにモジュール・ドキュメンテーションをレビューして下さい。



新規推移アクションの追加

全ての条件を設定後、変更を保存するために送信ボタンをクリックします。

次の推移アクションを作成してください:

- 「「管理」キューへプロセス・チケットを移動」(TA1)

  このアクションは推移「承認」(T1)が適用される時、実行されると思われます。

- 「チケットの責任者を「マネージャ」に変更」」 (TA2)

  推移「承認」(T1)の適用時、実行されるべきです。

- 「「従業員」キューへのプロセス・チケットの移動」(TA3)

  以下の場合実行されるべきです：

  - 推移:「否認」(T2)が適用された
  - 推移:「注文拒否」(T4) が適用された
  - 推移：「出荷受取」(T6)が適用された

- 「チケットの責任者を「従業員」へ変更」(TA4)

  以下の場合実行されるべきです：

  - 推移:「承認拒否」(T2)が適用された
  - 推移:「注文拒否」(T4) が適用された
  - 推移:「貨物受取」(T6) が」適用された

- 「プロセス・チケットを「購入」キュー」(TA5)へ移動する

  推移「承認済」(T3)が適用された場合に実行されるべきです。

- 「プロセス・チケットを「郵便局」キューへ移動」(TA6)

  推移「発注」(T5)が適用されたら実行されるべきです。

- 「チケットが正しく完了しました (TA7)」

  以下の場合実行されるべきです：

  - 推移:「貨物受取」(T6) が」適用された

- 「チケットが正しく完了しませんでした」(TA8)

  以下の場合実行されるべきです：

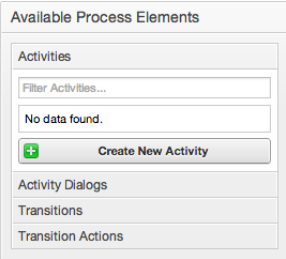
  - 推移:「承認拒否」(T2)が適用された
  - 推移:「注文拒否」(T4) が適用された

ご覧のように、同じ推移アクションが実行されるべき場所があります。したがって、それらを再使用することができるように推移アクションと推移とを自由にリンクすることを可能にすることは合理的なやり方です。

2.3.5. アクティビティ

アクティビティをひとつのバスケットとみなすアプローチを選択します。それは単数または複数のアクティビティ・ダイアログを含むことができます。

「利用可能なプロセス要素」中の「アクティビティ」をクリックして、次に、「新規アクティビティの作成」をクリックしてください。



図：新規アクティビティボタンの作成

開かれたポップ・アップ・画面中で「アクティビティ名」を記入して下さい。次に、必要なアクティビティ・ダイアログを"利用可能なアクティビティ・ダイアログ"プールからドラッグして、「割り当てされたアクティビティ・ダイアログ」プールへドロップします。このダイアログはチケット・ズーム画面で同じ順序で表示されるでしょう。その順序とは最上部から最下部、左から右へ推移するこの画面で定義されたものです。

この順序は特に最初のアクティビティで重要です。というのは、このアクティビティのための最初のアクティビティ・ダイアログはそれがプロセスが開始する時に表示されるただひとつのものだからです。

次のアクティビティを作成しましょう。

- 「依頼（要求）の記録」（A1)

  アクティビティ・ダイアログ「依頼（要求）の記録」を含みます（AD1）

- 「承認」(A2)

  アクティビティ・ダイアログ「承認」（AD3)と同様に「承認拒否」（AD2)を含みます。

- 「注文」（A3)

  アクティビティ・ダイアログ「発注」（AD5)と同様に「注文拒否」（AD4)を含みます。（AD5）

- 「入荷」(A4)

  アクティビティ・ダイアログ「貨物受取」（AD6)を含みます。

- 「プロセス完了」（A5）：これは可能なアクティビティ・ダイアログのないアクティビティです。「承認拒否」、「注文拒否」または「貨物受取」の後にセットされ、プロセスの終了を表します。

アクティビティが、プロセス・チケットの正確に定義された状態であることを明確に理解しました。成功した推移の後、プロセス・チケットはあるアクティビティから別のアクティビティへ遷移します。

2.3.6. 本の注文プロセスのパス

フローの記述者としてのプロセス、パズルの最後の欠けた 1 ピースで私たちの例を終えることにしましょう。私たちのケースでは、これは全注文ワークフローです。他のプロセスは事務用品注文あるいは完全に異なるプロセスでありえます。

プロセスには、スタート・アクティビティおよびスタート・アクティビティ・ダイアログをから成る、出発点があります。任意の新しい本オーダーについては、スタート・アクティビティ・ダイア

ログ(最初のアクティビティのための第1のアクティビティ・ダイアログ)は、表示される最初の画面です。これが完成し保存されれば、プロセス・チケットは作成され、形成されたワークフローに続くことができます。

プロセスには、プロセス・チケットどのようにしてがプロセスを通過できるのかについての指示も含まれています。これを「パス」と呼びましょう。それは、スタート・アクティビティ、1つ以上の推移(恐らく推移アクションを持った)および他の活動から成ります。

アクティビティがすでにアクティビティ・ダイアログを割り当てしたと仮定すると、アクティビティをアコーデオン（画面左部）からドラッグします。そしてそれをキャンバスエリア（以下のプロセス情報）にドロップします。プロセス・スタート(緑の円)からアクティビティまでの矢印が自動的に置かれることに注目してください。(これは最初のアクティビティです。また、その最初のアクティビティ・ダイアログは　プロセスが開始するときに表示される最初の画面です)。

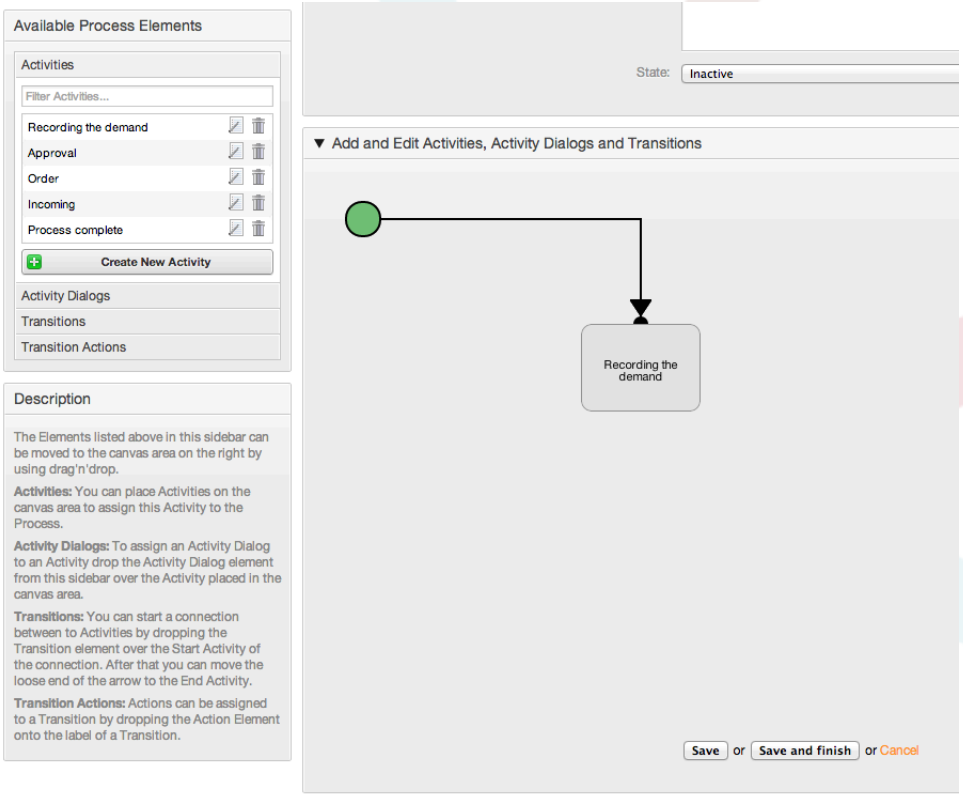
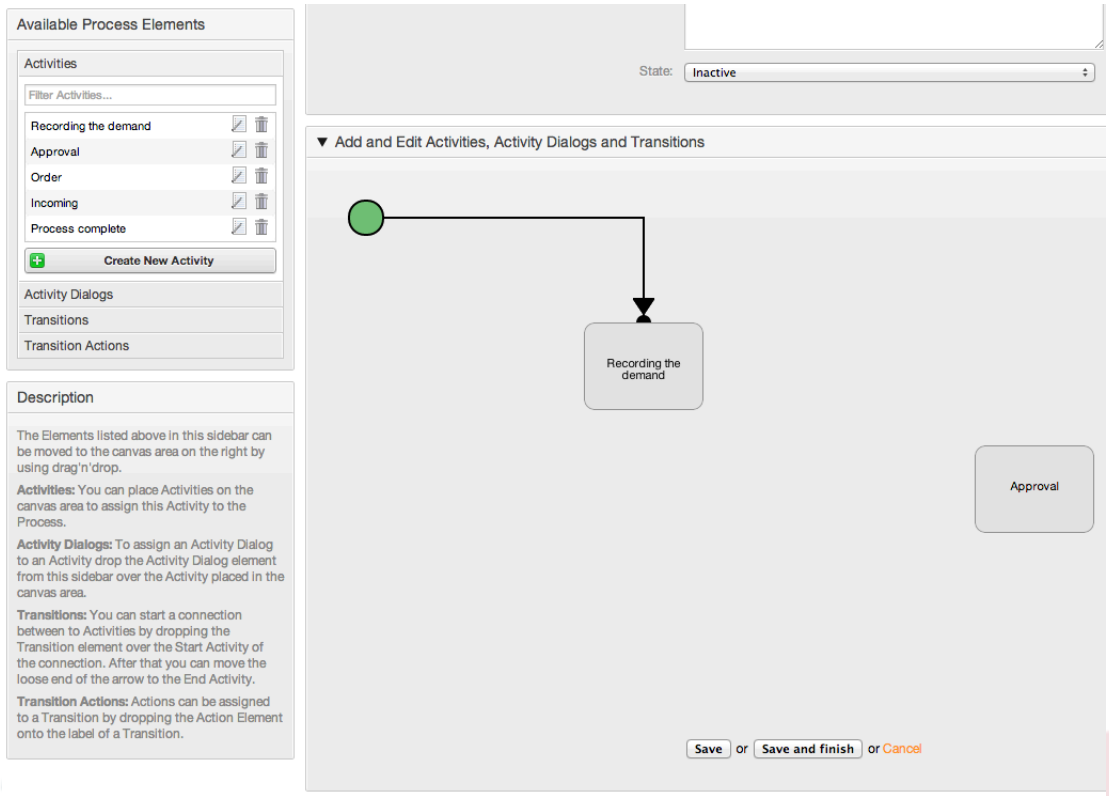


図:キャンバスに第1のアクティビティをドラッグしてください

次に、キャンバスに別のアクティビティをドラッグしてください。キャンバスに2つのアクティビティを持つことになります。ひとつはスタート点に接続され、もう一方は接続点を持ちません。それぞれのアクティビティ上でマウスポインタを持っていくと、それらアクティビティ・ダイアログを確認することができます。



図：第二のアクティビティをキャンバスにドラッグします

それからこの2つのアクティビティ間の「パス」を作成しましょう、そのために推移を使います。アコーディオン状の推移をクリックし、推移をドラッグしそれを最初のアクティビティの内部にドロップします。アクティビティはその色が
　推移が添付されたことを示して、変化することに気づいて下さい。推移が推移の端の矢印にドロップされるとすぐに、スタート開始点の隣に置かれるでしょう。
　アクティビティ間の接点を作成するために、推移の矢印の終了点をドラッグしてそれを他のアクティビティの内部にドロップします。

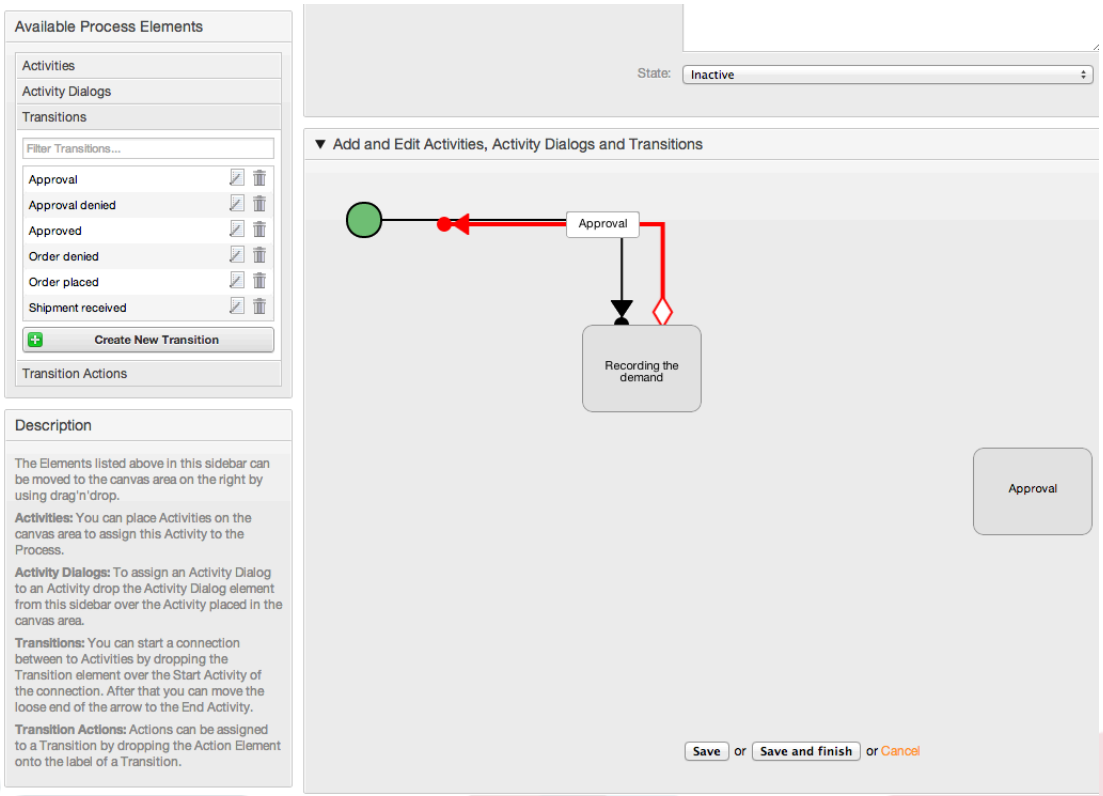


図:キャンバスに推移をドラッグします。

アクション間の「パス」が今定義されたので、次に推移に推移アクションを割り当てする必要があります。推移ラベル(キャンバス中の)をダブルクリックし、新しいポップアップ・ウィンドウを開きます。

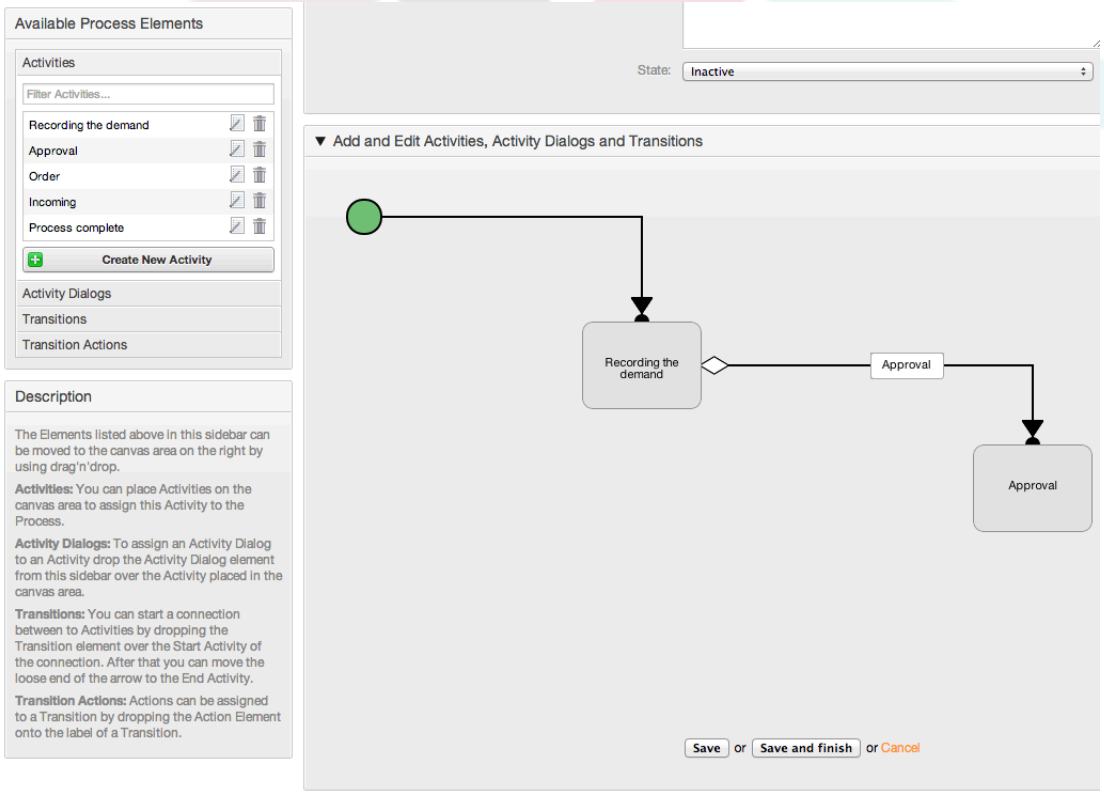


図:推移を使用して、アクティビティを接続します

必要な推移アクションを利用可能な推移アクションプールからドラッグし、それらを割り当てした推移アクションプールにドロップし、送信ボタンをクリックします。

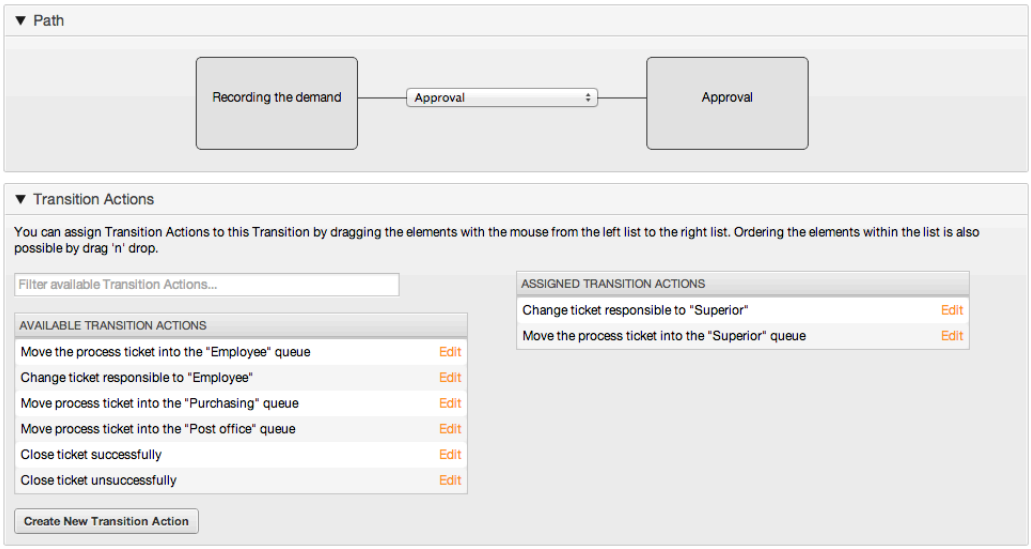


図:推移アクションを割り当てます

メインプロセス編集画面に戻って（キャンバス下にある他のすべての変更を保存するための）保存ボタンをクリックします。

次のアクティビティ、推移、推移アクションを追加することで「パス」を完成します。

「承認」まで依頼・要求を記録すること

- 出発点:アクティビティ:依頼・要求の記録(A1)
- 可能な推移:承認(T1)
  - この活動の条件が満たされれば、チケットはアクティビティに移ります:承認(A2)
  - さらに、次の推移アクションが実行されます:
    - 「「管理 」キューへチケットを移動させる」(TA1)
    - 「チケットの責任者を「マネージャー」に設定する」(TA2)

アクティビティ:「依頼・要求を記録する」（A1）が
プロセス・チケットの定義されたステップです。そのステップでは、推移：「承認」（T1)の可能性があります。これが適用されれば、チケットは次のアクティビティ:「承認」(A2)に移ります。 そして推移アクション:「「管理」キューへチケットを移動させる」(TA1)こと、及び「チケットの責任者として「マネージャー」を設定する」(TA2)が実行されます。アクティビティの中で:「承認」(A2)(アクティビティ・ダイアログ)、「承認拒否」(AD2)そして、「承認済」(AD3)が利用可能です。

承認

- 出発点:アクティビティ：「承認」(A2)
- 可能な推移:
  - 「承認拒否」(T2)
    - これが一致すれば、プロセス・チケットはアクティビティに移ります:「プロセス完了」(A5)
    - さらに、次の推移アクションが実行されます:

        - 「「従業員」キューへプロセス・チケットを移動」(TA3)
        - 「チケットの責任者を「従業員」設定」(TA4)
        - 「チケットが正しく完了しませんでした」(TA8)
    - 「承認済」(T3)
        - これが一致すれば、プロセス・チケットはアクティビティ：「注文」に移ります (A3)
        - さらに次の推移アクションが実行されます
            - 「プロセス・チケットを「購入」キューに移動します。」(TA5)

プロセスチケットのステップを定義する現在のアクティビティを見ればわかることがあります。推移には1個以上の可能性があり、それはまさにひとつのターゲット・アクティビティです。(そしておそらく1個以上の推移アクションです。)

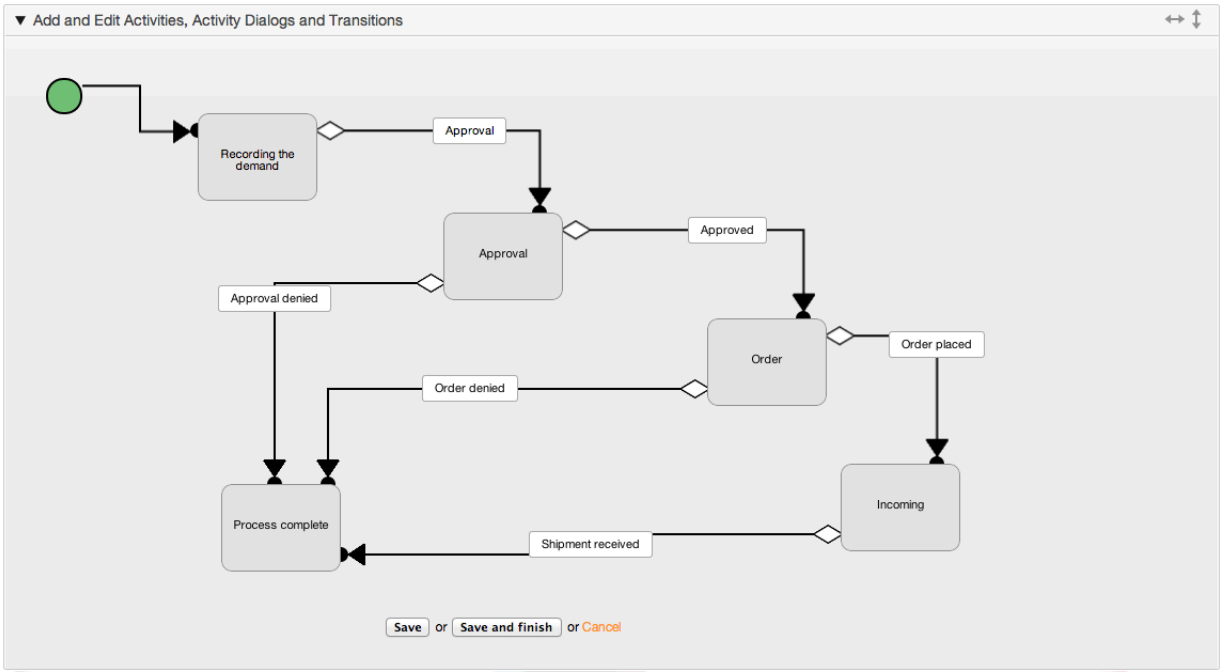
注文

- 開始点：アクティビティ「注文」(A3)
- 可能な推移:
    - 「注文拒否」(T4)
        - これが一致すれば、プロセス・チケットはアクティビティに移ります:「プロセス完了」(A5)
        - さらに、次の推移アクションが実行されます:
            - 「「従業員」キューへプロセス・チケットを移動」(TA3)
            - 「チケットの責任者を「従業員」設定」(TA4)
            - 「チケットが正しく完了しませんでした」(TA8)
    - 「発注」(T5)
        - これが一致すれば、プロセス・チケットはアクティビティに移るでしょう:「入信」(A5)
        - さらに次の推移アクションが実行されます
            - 「「郵便局」キューにプロセス・チケットを移動させてください」(TA6)

Incoming (入信)

- 出発点：アクティビティ「入信」A4
- 可能な推移:
    - 「貨物受取」(T6)
        - これが一致すれば、プロセス・チケットはアクティビティに移ります:「プロセス完了」(A5)
        - さらに、次の推移アクションが実行されます:
            - 「「従業員」キューへプロセス・チケットを移動」(TA3)
            - 「チケットの責任者を「従業員」設定」(TA4)
            - 「チケットが正しく完了しました (TA7)」

本の注文プロセスの完全なパスはこのように見えるでしょう。



図：本の注文の完全なプロセス・パス

プロセス・パスを終了した後、キャンバスの下方部分中の「セーブする」ボタンをクリックして、次に、「全てのプロセスを同期」ボタンをクリックしてください。これはデータベースからすべてのプロセス情報を収集しキャッシュ・ファイル(Perl言語で)を作成します。このキャッシュ・ファイルは実際、プロセス・チケットを作成するか使用するためにシステムが使用するプロセス・構成です。

プロセス(GUIの中の)に行なわれるどんな変更も、システムで変更を反映するために再度キャッシュ・ファイルを同期させる必要があります。

さらに、YAMLファイルから全工程をインポートすることは可能です。しかし、インポートの前に各プロセスによって必要な動的フィールド、ユーザ、キューなどをすべて作ることがまだ必要です。

プロセスがACLの使用を要求する場合、手動でセットされることも必要であることに注目してください。

以下は本の注文の処理例のための完全なYAMLファイルです:

```
---
Activities:
  A1:
    ActivityDialogs:
      - AD1
    ChangeTime: 2012-11-23 14:49:22
    Config:
      ActivityDialog:
        1: AD1
    CreateTime: 2012-11-23 11:49:38
    EntityID: A1
    ID: 151
    Name: Recording the demand
  A2:
    ActivityDialogs:
      - AD2
      - AD3
    ChangeTime: 2012-12-13 00:55:12
    Config:
```

```
      ActivityDialog:
        1: AD2
        2: AD3
    CreateTime: 2012-11-23 11:50:11
    EntityID: A2
    ID: 152
    Name: Approval
  A3:
    ActivityDialogs:
      - AD4
      - AD5
    ChangeTime: 2012-11-23 18:12:14
    Config:
      ActivityDialog:
        1: AD4
        2: AD5
    CreateTime: 2012-11-23 11:50:35
    EntityID: A3
    ID: 153
    Name: Order
  A4:
    ActivityDialogs:
      - AD6
    ChangeTime: 2012-11-23 18:12:35
    Config:
      ActivityDialog:
        1: AD6
    CreateTime: 2012-11-23 11:51:00
    EntityID: A4
    ID: 154
    Name: Incoming
  A5:
    ActivityDialogs: []
    ChangeTime: 2012-11-23 11:51:33
    Config: {}
    CreateTime: 2012-11-23 11:51:33
    EntityID: A5
    ID: 155
    Name: Process complete
ActivityDialogs:
  AD1:
    ChangeTime: 2012-12-06 02:16:21
    Config:
      DescriptionLong: ''
      DescriptionShort: Recoding the demand
      FieldOrder:
        - DynamicField_Author
        - DynamicField_ISBN
        - DynamicField_Title
        - DynamicField_Status
      Fields:
        DynamicField_Author:
          DefaultValue: ''
          DescriptionLong: ''
          DescriptionShort: ''
          Display: 1
        DynamicField_ISBN:
          DefaultValue: ''
          DescriptionLong: ''
          DescriptionShort: ''
          Display: 1
        DynamicField_Status:
          DefaultValue: ''
          DescriptionLong: ''
          DescriptionShort: ''
          Display: 1
        DynamicField_Title:
          DefaultValue: ''
          DescriptionLong: ''
          DescriptionShort: ''
          Display: 1
      Interface:
```

```
      - AgentInterface
    Permission: ''
    RequiredLock: 0
    SubmitAdviceText: ''
    SubmitButtonText: ''
  CreateTime: 2012-11-23 14:34:43
  EntityID: AD1
  ID: 154
  Name: Recording the demand
AD2:
  ChangeTime: 2012-11-23 14:57:41
  Config:
    DescriptionLong: ''
    DescriptionShort: Approval denied
    FieldOrder:
      - Article
      - DynamicField_Status
    Fields:
      Article:
        Config:
          ArticleType: note-internal
        DefaultValue: ''
        DescriptionLong: ''
        DescriptionShort: ''
        Display: 1
      DynamicField_Status:
        DefaultValue: ''
        DescriptionLong: ''
        DescriptionShort: ''
        Display: 1
    Interface:
      - AgentInterface
    Permission: ''
    RequiredLock: 0
    SubmitAdviceText: ''
    SubmitButtonText: Deny Request
  CreateTime: 2012-11-23 14:36:39
  EntityID: AD2
  ID: 155
  Name: Approval denied
AD3:
  ChangeTime: 2012-12-14 03:14:23
  Config:
    DescriptionLong: ''
    DescriptionShort: Approved
    FieldOrder:
      - DynamicField_Status
    Fields:
      DynamicField_Status:
        DefaultValue: ''
        DescriptionLong: ''
        DescriptionShort: ''
        Display: 1
    Interface:
      - AgentInterface
    Permission: ''
    RequiredLock: 0
    SubmitAdviceText: ''
    SubmitButtonText: Approve Request
  CreateTime: 2012-11-23 14:37:35
  EntityID: AD3
  ID: 156
  Name: Approved
AD4:
  ChangeTime: 2012-11-23 14:58:52
  Config:
    DescriptionLong: ''
    DescriptionShort: Order rejected
    FieldOrder:
      - Article
      - DynamicField_Status
    Fields:
```

```
      Article:
        Config:
          ArticleType: note-internal
        DefaultValue: ''
        DescriptionLong: ''
        DescriptionShort: ''
        Display: 1
      DynamicField_Status:
        DefaultValue: ''
        DescriptionLong: ''
        DescriptionShort: ''
        Display: 1
    Interface:
      - AgentInterface
    Permission: ''
    RequiredLock: 0
    SubmitAdviceText: ''
    SubmitButtonText: Reject Order
  CreateTime: 2012-11-23 14:38:48
  EntityID: AD4
  ID: 157
  Name: Order rejected
AD5:
  ChangeTime: 2012-12-06 02:20:12
  Config:
    DescriptionLong: ''
    DescriptionShort: Order placed
    FieldOrder:
      - DynamicField_DeliveryDate
      - DynamicField_Price
      - DynamicField_Supplier
      - DynamicField_Status
    Fields:
      DynamicField_DeliveryDate:
        DefaultValue: ''
        DescriptionLong: ''
        DescriptionShort: ''
        Display: 1
      DynamicField_Price:
        DefaultValue: ''
        DescriptionLong: ''
        DescriptionShort: ''
        Display: 1
      DynamicField_Status:
        DefaultValue: ''
        DescriptionLong: ''
        DescriptionShort: ''
        Display: 1
      DynamicField_Supplier:
        DefaultValue: ''
        DescriptionLong: ''
        DescriptionShort: ''
        Display: 1
    Interface:
      - AgentInterface
    Permission: ''
    RequiredLock: 0
    SubmitAdviceText: ''
    SubmitButtonText: Place Order
  CreateTime: 2012-11-23 14:41:28
  EntityID: AD5
  ID: 158
  Name: Order placed
AD6:
  ChangeTime: 2012-11-23 14:42:43
  Config:
    DescriptionLong: ''
    DescriptionShort: Shipment received
    FieldOrder:
      - DynamicField_DateOfReceipt
      - DynamicField_Status
    Fields:
```

```
      DynamicField_DateOfReceipt:
        DefaultValue: ''
        DescriptionLong: ''
        DescriptionShort: ''
        Display: 1
      DynamicField_Status:
        DefaultValue: ''
        DescriptionLong: ''
        DescriptionShort: ''
        Display: 1
    Interface:
      - AgentInterface
    Permission: ''
    RequiredLock: 0
    SubmitAdviceText: ''
    SubmitButtonText: ''
  CreateTime: 2012-11-23 14:42:43
  EntityID: AD6
  ID: 159
  Name: Shipment received
Process:
  Activities:
    - A1
    - A2
    - A3
    - A4
    - A5
  ChangeTime: 2012-12-06 02:31:59
  Config:
    Description: The process to order a book
    Path:
      A1:
        T1:
          ActivityEntityID: A2
          TransitionAction:
            - TA2
            - TA1
      A2:
        T2:
          ActivityEntityID: A5
          TransitionAction:
            - TA3
            - TA4
            - TA8
        T3:
          ActivityEntityID: A3
          TransitionAction:
            - TA5
      A3:
        T4:
          ActivityEntityID: A5
          TransitionAction:
            - TA3
            - TA4
            - TA8
        T5:
          ActivityEntityID: A4
          TransitionAction:
            - TA6
      A4:
        T6:
          ActivityEntityID: A5
          TransitionAction:
            - TA3
            - TA4
            - TA7
      A5: {}
    StartActivity: A1
    StartActivityDialog: AD1
  CreateTime: 2012-11-23 11:45:12
  EntityID: P1
  ID: 94
```

```
  Layout:
    A1:
      left: 172
      top: 63
    A2:
      left: 402
      top: 156
    A3:
      left: 649
      top: 255
    A4:
      left: 774
      top: 391
    A5:
      left: 194
      top: 410
  Name: Book ordering
  State: Active
  StateEntityID: S1
  TransitionActions:
    - TA1
    - TA2
    - TA3
    - TA4
    - TA8
    - TA5
    - TA3
    - TA4
    - TA8
    - TA6
    - TA3
    - TA4
    - TA7
  Transitions:
    - T1
    - T2
    - T3
    - T4
    - T5
    - T6
TransitionActions:
  TA1:
    ChangeTime: 2012-11-23 16:01:37
    Config:
      Config:
        Queue: Management
      Module: Kernel::System::ProcessManagement::TransitionAction::TicketQueueSet
    CreateTime: 2012-11-23 15:50:59
    EntityID: TA1
    ID: 61
    Name: Move the process ticket into the "Management" queue
  TA2:
    ChangeTime: 2012-11-23 16:02:12
    Config:
      Config:
        Responsible: manager
      Module: Kernel::System::ProcessManagement::TransitionAction::TicketResponsibleSet
    CreateTime: 2012-11-23 15:58:22
    EntityID: TA2
    ID: 62
    Name: Change ticket responsible to "manager"
  TA3:
    ChangeTime: 2012-11-24 14:27:02
    Config:
      Config:
        Queue: Employees
      Module: Kernel::System::ProcessManagement::TransitionAction::TicketQueueSet
    CreateTime: 2012-11-23 16:02:54
    EntityID: TA3
    ID: 63
    Name: Move the process ticket into the "Employees" queue
  TA4:
```

```
    ChangeTime: 2012-11-23 16:04:06
    Config:
      Config:
        Responsible: Employee
      Module: Kernel::System::ProcessManagement::TransitionAction::TicketResponsibleSet
    CreateTime: 2012-11-23 16:04:06
    EntityID: TA4
    ID: 64
    Name: Change ticket responsible to "Employee"
  TA5:
    ChangeTime: 2012-12-06 02:18:34
    Config:
      Config:
        Queue: Purchasing
      Module: Kernel::System::ProcessManagement::TransitionAction::TicketQueueSet
    CreateTime: 2012-11-23 16:04:54
    EntityID: TA5
    ID: 65
    Name: Move process ticket into the "Purchasing" queue
  TA6:
    ChangeTime: 2012-12-06 02:18:48
    Config:
      Config:
        Queue: Post office
      Module: Kernel::System::ProcessManagement::TransitionAction::TicketQueueSet
    CreateTime: 2012-11-23 16:06:20
    EntityID: TA6
    ID: 66
    Name: Move process ticket into the "Post office" queue
  TA7:
    ChangeTime: 2012-12-06 02:29:55
    Config:
      Config:
        State: closed successful
      Module: Kernel::System::ProcessManagement::TransitionAction::TicketStateSet
    CreateTime: 2012-12-06 02:29:27
    EntityID: TA7
    ID: 67
    Name: Close ticket successfully
  TA8:
    ChangeTime: 2012-12-06 02:31:12
    Config:
      Config:
        State: closed unsuccessful
      Module: Kernel::System::ProcessManagement::TransitionAction::TicketStateSet
    CreateTime: 2012-12-06 02:31:12
    EntityID: TA8
    ID: 68
    Name: Close ticket unsuccessfully
Transitions:
  T1:
    ChangeTime: 2012-11-23 15:12:20
    Config:
      Condition:
        1:
          Fields:
            DynamicField_Status:
              Match: Approval
              Type: String
          Type: and
      ConditionLinking: and
    CreateTime: 2012-11-23 11:53:52
    EntityID: T1
    ID: 94
    Name: Approval
  T2:
    ChangeTime: 2012-11-23 15:12:50
    Config:
      Condition:
        1:
          Fields:
            DynamicField_Status:
```

```
            Match: Approval denied
            Type: String
        Type: and
    ConditionLinking: and
  CreateTime: 2012-11-23 11:54:26
  EntityID: T2
  ID: 95
  Name: Approval denied
T3:
  ChangeTime: 2012-11-23 15:13:29
  Config:
    Condition:
      1:
        Fields:
          DynamicField_Status:
            Match: Approved
            Type: String
        Type: and
    ConditionLinking: and
  CreateTime: 2012-11-23 11:54:54
  EntityID: T3
  ID: 96
  Name: Approved
T4:
  ChangeTime: 2012-11-23 15:14:08
  Config:
    Condition:
      1:
        Fields:
          DynamicField_Status:
            Match: Order denied
            Type: String
        Type: and
    ConditionLinking: and
  CreateTime: 2012-11-23 11:55:25
  EntityID: T4
  ID: 97
  Name: Order denied
T5:
  ChangeTime: 2012-11-23 18:30:33
  Config:
    Condition:
      1:
        Fields:
          DynamicField_Status:
            Match: Order placed
            Type: String
        Type: and
    ConditionLinking: and
  CreateTime: 2012-11-23 11:56:15
  EntityID: T5
  ID: 98
  Name: Order placed
T6:
  ChangeTime: 2012-11-23 15:15:30
  Config:
    Condition:
      1:
        Fields:
          DynamicField_Status:
            Match: Shipment received
            Type: String
        Type: and
    ConditionLinking: and
  CreateTime: 2012-11-23 11:56:48
  EntityID: T6
  ID: 99
  Name: Shipment received
```

2.4. プロセス構成リファレンス

2.4.1. プロセス

1つのプロセスはワークフロー/プロセスのパスを形成します。このパス上の中間地点はアクティビティまたは推移になりえます。これらについては後程お話します。

2.4.1.1. プロセス構成

プロセス構成は、ファイル　Kernel/Config.pm中で行われます。しかし　Kernel/Config/Files/MyProcess.pm　のような新しいファイルを作成することが強く推奨されます。GUIがファイル Kernel/Config/File/ZZZProcessManagement を生成することに気づいて下さい。そのファイル名を使用することは回避してください。そうでなければ、同期プロセスの際ににそれは上書きされるでしょう。例：プロセス構成(プロセス・キャッシュ・ファイルからの)を見ましょう:

```
$Self->{'Process'} = {
    'P1' => {
        Name                => '本の注文',
        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 注文',
        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

プロセスの名前、新しいプロセス・チケットを作成する場合、担当者はこれを選択することができます。

2.4.1.3. CreateTime

プロセスの作成時刻。

2.4.1.4. CreateBy

プロセスを作成するユーザのUID

2.4.1.5. ChangeTime

プロセスの変更時刻

2.4.1.6. ChangeBy

プロセスの最終変更ユーザのUID

2.4.1.7. State

プロセスの状態を定義します。可能な値:

- 「Active（アクティブ）」は、新しいプロセス・チケットの中で使用することができる全てのプロセスです。
- 「FadeAway（消失）」は、
  もはやさらなる新規チケットを選択できないプロセスです。既存チケットはまだプロセスを使用可能です。
- 「Inactive」（インアクティブ）は、不活性化されたプロセスです。新規も既存チケットも使用できません。

2.4.1.8. StartActivity

新しいプロセス・チケットを作成する場合、StartActivityを定義しなければいけません。チケットが作成されるとすぐに、このアクティビティは最初の推移チェック用の基礎としてセットされ使用されるでしょう。

2.4.1.9. StartActivityDialog

新しいプロセス・チケットでは、StartActivityDialogは定義されなければいけません。新しいプロセス・チケット(プロセスが選択された後)を作成する時、これが示されるでしょう。このポイントでは、チケットはまだ存在しません、それはStartActivityDialogを提出した後に作成されるでしょう。

2.4.1.10. パス

パスは、現在のプロセスのために、アクティビティの構造およびそれらの間の可能な推移を含みます。そしてさらに発生する推移アクションは移行時に起こります。これは、プロセス・チケットがとることができる方法をコントロールします。

```
'A1' => {
    'T1' => {
        ActivityEntityID => 'A2',
    },
    'T2' => {
        ActivityEntityID => 'A3',
    },
    'T3' => {
        ActivityEntityID => 'A4',
        TransitionAction => ['TA1', 'TA2'],
    },
},
```

プロセス・チケットがアクティビティ('A1')にある場合、それには別の活動に到着する3つの可能な方法がある。推移(「T1」)に、「T3」に、条件は定義され、プロセス・チケットは別のアクティビティへの遷移を完了しなければならない。

この場合、推移「T2」に必要とされるプロセス・チケットおよびその動的なフィールドの値がすべて正確ならば、チケットは活動「A1」から「A3」に移動されるだろう。ActivityDialogが提出されるか、あるいはチケットに他の変更が行なわれる後、それは現在のアクティビティからの可能な推移をチェックされるだろう。複数の推移が可能ならば、最初のものが使用されるだろう (TransitionIDsをソートする数値ソートに基づく)。

さらに、パス設定中の推移に推移アクションを割り当てることは可能です。これらは成功した推移の後に実行されるモジュールである。それらは例でのような配列形式で指定しなければならない。詳細は後述します。

2.4.2. アクティビティ

アクティビティは1つ以上のアクティビティ・ダイアログを含んでおり、プロセスの「ステップ」をモデル化します。現在のアクティビティの全てのアクティビティダイアログはチケット・ズームで表示され、推移の条件が満たされるまで使用することができます。

2.4.2.1. アクティビティ構成

サンプルのアクティビティ構成を見ましょう:

```
$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

アクティビティの名前

2.4.2.3. CreateTime

アクティビティの作成時刻

2.4.2.4. CreateBy

アクティビティの作成ユーザーのユーザのUID

2.4.2.5. ChangeTime

アクティビティの変更時刻

2.4.2.6. ChangeBy

アクティビティの最終変更ユーザのユーザのUID

2.4.2.7. ActivityDialog (アクティビティ・ダイアログ)

アクティビティダイアログは、このアクティビティにおいて利用可能なアクティビティ・ダイアログのリストを含んでいます。現在のアクティビティの全てのアクティビティ・ダイアログはチケット・ズームで表示されます。それらの順序は、構成の中の順序でここでセットされます「AD5」は「AD6」および「AD1」の前に示されます。

2.4.3. ActivityDialog (アクティビティ・ダイアログ)

アクティビティ・ダイアログは特別の画面で、異なるアクティビティの中で使用することができます。

2.4.3.1. ActivityDialog (アクティビティ・ダイアログ) の構成

サンプルのコンフィグを見ましょう。

```
$Self->{'Process::ActivityDialog'} = {
    'AD1' => {
        Name             => 'ActivityDialog 1 optional',
        DescriptionShort => '基本情報',
        DescriptionLong  => 'IT注文に関する必要な基本情報を挿入して下さい。',
        CreateTime       => '28-02-2012 13:37:00',
        CreateBy         => '1',
        ChangeTime       => '29-02-2012 13:37:00',
        ChangeBy         => '1',
        Fields => {
            PriorityID => {
                DescriptionShort => '優先度 ID',
                DescriptionLong  => '優先度をここに入力して下さい。',
                Display          => 2,
            },
        },
        FieldOrder       => [ 'PriorityID' ],
        SubmitAdviceText => 'ノート：フォームを送信したい場合は...',
        SubmitButtonText => '要求を送信',
    },
    'AD2' => {
        Name => 'ActivityDialog 2 optional',
        DescriptionShort => '基本情報',
        DescriptionLong  => '本の注文に関する必要な基本情報を挿入して下さい。',
        CreateTime       => '28-02-2012 13:37:00',
        CreateBy         => '1',
        ChangeTime       => '29-02-2012 13:37:00',
        ChangeBy         => '1',
        Fields => {
            StateID => {
                DescriptionShort => '状態 ID',
                DescriptionLong  => '状態をここに入力して下さい。',
                Display          => 2,
                DefaultValue     => '2',
            },
            Queue => {
                DescriptionShort => 'キュー ID',
                DescriptionLong  => 'キューをここに入力して下さい。',
                Display          => 2,
                DefaultValue     => 'Raw',
            },
            Title => {
                DescriptionShort => 'タイトル',
                DescriptionLong  => 'タイトルをここに入力して下さい。',
                Display          => 1,
                DefaultValue     => 'Default Title',
            },
```

```
            DynamicField_Anzahl => {
                DescriptionShort    => '金額',
                DescriptionLong     => '金額をここに入力して下さい。',
                Display             => 2,
                DefaultValue        => '4',
            },
        },
        FieldOrder       => [ 'DynamicField_Anzahl', 'StateID', 'Queue', 'Title' ],
        SubmitAdviceText => 'ノート：フォームを送信する場合…',
        SubmitButtonText => '要求を送信',
    },
};
```

2.4.3.2. Name

アクティビティ・ダイアログの名前

2.4.3.3. CreateTime

アクティビティ・ダイアログ作成時刻

2.4.3.4. CreateBy

このアクティビティ・ダイアログを作成したユーザのUID

2.4.3.5. ChangeTime

アクティビティ・ダイアログの最終変更時刻

2.4.3.6. ChangeBy

このアクティビティダイアログの最終変更者のユーザのUID

2.4.3.7. フィールド

このアクティビティダイアログで表示することができる全てのフィールドを含んでいます。次のフィールドは現在使用することができます:

```
    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  # 全ての動的フィールド用
```

単一のフィールド構成の例：

```
StateID => {
    DescriptionShort => 'State ID',
    DescriptionLong  => 'Enter the state here',
    Display          => 2,
    DefaultValue     => '2',
},
```

フィールド「記事」は特殊なケースです。それが「フィールド」構成の中にあれば、アクティビティ・ダイアログは件名フィールドおよび添付ファイル取扱い可能な機能を持った完全なリッチテキスト・エディターを内蔵します。その後、入力されたテキストは、記事としてチケットに加えられ、電子メールで送られます。サンプルの記事フィールド構成を見ましょう:

```
Article => {
    DescriptionShort => 'ここにコメントを挿入して下さい',
    DescriptionLong => '',
    Display         => 1,
    Config          => {
        ArticleType  => 'note-internal',
        LabelSubject => '',
        LabelBody    => '',
    },
},
```

フィールド構成オプションを見ましょう:

2.4.3.7.1. DescriptionShort (短い摘要)

フィールドタイトルで表示される短い摘要 (オプション) 。

2.4.3.7.2. DescriptionLong (長い摘要)

マウスがフィールド上に乗る時、表示されるより長い摘要 (オプション) 。フィールドの入力方法についての例示アドバイス。

2.4.3.7.3. Display (表示)

フィールドが示される　かつ、または　必須の場合のコントロール。可能な値:

- '0': フィールドは不可視です。フィールド値が自動的にセットならより便利でしょう。構成されたDefaultValueはこのケースでは格納されるでしょう。
- '1' :フィールドは可視ですが任意です。
- '2': fフィールドは可視でかつ必須です。次のフィールドが単に不可視にも必須にもなりえます:

  ```
  QueueID
  Queue
  State
  StateID
  Lock
  LockID
  Priority
  PriorityID
  Type
  TypeID
  ```

フィールドが任意として設定され、ユーザが値を送信しなければ、アクティビティ・ダイアログがユーザによって送信される時、デフォルト値が保存されます。

2.4.3.7.4. DefaultValue (デフォルト値)

「ID'」(QueueID、OwnerIDのように)を備えたフィールドに関して、これは、値のデータベースIDを参照します。'ID'(キュー、オーナーのように)のない他のフィールドについては、DefaultValueは値自体を含んでいなければいけません。例:

```
Queue => {
    DescriptionShort => 'Queue',
    DescriptionLong  => 'Enter the queue here',
    Display          => 2,
    DefaultValue     => 'Raw',
},
```

2.4.3.8. FieldOrder (フィールド順序)

ここで、フィールドの表示順序は設定されます。重要:保存する時単に設定されたフィールドが考慮されるので、目に見えないフィールドはさらにここで設定されなければいけません。設定されないフィールドは保存されないでしょう。

2.4.3.9. SubmitAdviceText (アドバイス用テキストの送信)

補足的ヘルプかアドバイス用テキストのための送信ボタン上に示される任意のテキスト。

2.4.3.10. SubmitButtonText (送信ボタンのテキスト)

送信ボタンのための任意のカスタム・テキスト。

2.4.4. Transition (推移)

推移は、設定可能な条件に基づいて決定します。プロセス中でパスが取得されているような条件です。。(例えば　プロセス・チケットが移動されうるアクティビティへのパスです)

2.4.4.1. Transition configuration (推移構成)

サンプルを見ましょう:

```
$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

推移の名前。

2.4.4.3. CreateTime

アクティビティ・ダイアログ作成時刻

2.4.4.4. CreateBy

この推移を作成したユーザのUID。

2.4.4.5. ChangeTime

アクティビティ・ダイアログの最終変更時刻

2.4.4.6. ChangeBy

この推移の最終変更者のユーザのUID。

2.4.4.7. Condition (条件)

この推移が効果を現わす必須条件をすべて含んでいます。例:

```
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',
            },
        },
    },
},
```

構成の条件を詳細に見ましょう。

2.4.4.7.1. タイプ (条件)

異なる条件要素が互いに接続される方法を指定します。可能な値:

- 'and':　これはデフォルトです。推移が有効になるために条件をすべて満たさなければなりません。

- 'or': 少なくとも1つの条件は一致しなければいけません。
- 'xor': 1つの条件のみ一致しなければいけません。

2.4.4.7.2. Cond1

これはサンプルの条件の名前です。それは自由に選ぶことができます。条件はソート順に評価されます。

2.4.4.7.3. タイプ(Cond)

この条件の個々のフィールド試験が互いに接続される方法を指定します。可能な値:

- 'and': これはデフォルトです。全てのフィールド・テスト条件をすべて満たさなければなりません。
- 'or': 少なくとも1つの条件は一致しなければいけません。
- 'xor': 1つの条件のみ一致しなければいけません。

2.4.4.7.4. フィールド

その値がテストされるべき特別のフィールドを指定します。私たちの例から:

```
Fields => {
    StateID => {
        Type  => 'String',
        Match => '1',
    },
```

2.4.4.7.5. StateID

フィールド名の例。次のチケット・フィールドが使用されます。:

```
    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
```

「ID」(SLAIDのように)を備えたフィールドをテストする時、フィールドのデータベースIDは、実際の値がそのために使用される他のフィールド(SLAのような)のための試験に使用されます。

2.4.4.7.6. タイプ

フィールド試験の種類を決定します。可能な値:

- 「String」: フィールド値を「Match」で指定されたStringと比較します。それらが正確に同じである場合マッチします。
- 「ハッシュ(Hush)」:フィールド値(ハッシュ)を「Match」で指定されたハッシュと比較します。ハッシュ値はすべて同じでなければいけない。
- 「配列(array)」:フィールド値(配列)を「Match」で指定された配列と比較します。両方のリストは同じでなければならない。
- 「Regex:」:フィールド値は正規表現でテストすることができます。「Match」が基本的条件として*qr{}xms*を含むことは重要です。{}間に、実際の正規表現を記述することができます。
- 「モジュール:」:条件チェックのためにperlモジュールを使用することができます。チェックが1を返すならば、チェックは成功です。`Kernel/System/ProcessManagement/TransitionValidation/ValidateDemo.pm`. でサンプル・モジュールを見つけることができます。

2.4.5. 推移アクション

推移アクションは、推移がうまく適用された後で、きっかけとなるアクションです。(プロセス・チケットがひとつのアクティビティから別のアクティビティへ移る場合)これらの推移アクションはチケットで異なる変更を実施するために使われます。(例えば、キューまたはチケットのオーナーの変更)そうすれば、さらに、他の複雑な変更を行なうために自分の推移アクションを作成することができます。

2.4.5.1. 推移アクション構成

例を見ましょう::

```
$Self->{'Process::TransitionAction'} = {
    'TA1' => {
        Name   => 'Queue Move',
        Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketQueueSet',
        Config => {
            Queue  => 'Junk',
            UserID => 123,
        },
    },
};
```

2.4.5.2. Name

推移アクションの名前。

2.4.5.3. Module(モジュール)

使用されるパール・モジュールを指定します。

2.4.5.4. Config(構成)

このパラメーターは、モジュールに必要なセッティングをすべて含んでいます。その内容は、使用される特定の推移アクション・モジュールに依存します。詳細については、個々のモジュールのドキュメンテーションを参照してください。私たちの例において、キューだけが指定されねばなりません。

推移アクションの「構成」パラメタ内部のUserIDの使用は、全ての推移アクション(OTRS 3.2.4以来の)によって認めてられています。この例では、推移を引き起こすユーザーが、チケットを「ジャンク」キューへ移動する権限を持っているかどうかが、特に重要になりえます。ユーザーID 123を持つユーザーが持っているかもしれませんが。。

2.4.5.5. 推移アクション・モジュールの再使用

推移アクション・モジュールを複数回使用するためには、単に構成中でいくつか TransitionActionsを指定してください。例:

```
    $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',
            },
        },
    };
```

ここで、同じモジュールは、「未対応」キューにプロセス・チケット移動するために、そしてある時はジャンクキューへそれを移動するために、使用されます。特別の推移に使用されることが求められる推移アクションは、プロセス構成の「パス」設定から決定されます。

2.4.5.6. 利用可能な推移アクション

OTRSは、あなたのプロセスで使用できるいくつかの推移アクションを搭載しています。ここでは、それらのドキュメンテーション、およびどのようにしてそれらを設定する必要があるかを見つけることができます。

2.4.5.6.1. DynamicFieldSet (動的フィールドのセット)

プロセス・チケットで1つ以上の動的フィールドをセットします。

```
$Self->{'Process::TransitionAction'} = {
    'TA1' => {
        Name   => 'Set DynamicField Master to Master and Approved to 1',
        Module => 'Kernel::System::ProcessManagement::TransitionAction::DynamicFieldSet',
        Config => {
            MasterSlave => 'Master',
            Approved    => '1',

        },
    },
};
```

「Name」は、設定された推移アクション名前を指定します。

「MasterSlave」と「Approved：承認済」は DynamicField名のサンプルを付与されます。フィールド値 (「Master'」and 「1」) はこの推移アクションによってセットされます。

2.4.5.6.2. TicketArticleCreate (チケット記事の作成)

記事を作成し、ノートあるいは電子メール返答を作成するために使用することができます。例:

```
$Self->{'Process::TransitionAction'} = {
    'TA1' => {
        Name   => 'Article Create Note Internal',
        Module =>
 'Kernel::System::ProcessManagement::TransitionAction::TicketArticleCreate',
        Config => {
```

```
            ArticleType      => 'note-internal',                                    #
note-external|phone|fax|sms|...
                                                                                    #
excluding any email type
            SenderType       => 'agent',                                            #
agent|system|customer
            ContentType      => 'text/plain; charset=ISO-8859-15',                  # or
optional Charset & MimeType
            Subject          => 'some short description',                           #
required
            Body             => 'the message text',                                 #
required
            HistoryType      => 'OwnerUpdate',                                      #
EmailCustomer|Move|AddNote|PriorityUpdate|WebRequestCustomer|...
            HistoryComment   => 'Some free text!',
            From             => 'Some Agent <email@example.com>',                   #
not required but useful
            To               => 'Some Customer A <customer-a@example.com>',         #
not required but useful
            Cc               => 'Some Customer B <customer-b@example.com>',         #
not required but useful
            ReplyTo          => 'Some Customer B <customer-b@example.com>',         #
not required
            InReplyTo        => '<asdasdasd.12@example.com>',                       #
not required but useful
            References       => '<asdasdasd.1@example.com> <asdasdasd.12@example.com>', #
not required but useful
            NoAgentNotify    => 0,                                                  # if
you don't want to send agent notifications
            AutoResponseType => 'auto reply',                                       #
auto reject|auto follow up|auto reply/new ticket|auto remove

            ForceNotificationToUserID   => [ 1, 43, 56 ],                           # if
you want to force somebody
            ExcludeNotificationToUserID => [ 43, 56 ],
                 # if you want full exclude somebody from notfications,
                 # will also be removed in To: line of article,
                 # higher prio as ForceNotificationToUserID
            ExcludeMuteNotificationToUserID => [ 43, 56 ],
                 # the same as ExcludeNotificationToUserID but only the
                 # sending gets muted, agent will still shown in To:
                 # line of article
        },
    },
};
```

「Name」は、設定された推移アクション名前を指定します。それは自由に選ぶことができますが、設定されたアクションの目的を反映するべきです。

「ArticleType」は、作成される記事のタイプを定義します。可能な値:電話、ファックス、sms、webrequest、ノート内部、ノート外部およびノート報告書。

「SenderType」は送り手タイプの記事を定義します。可能な値:担当者、システム、顧客。

「ContentType」は、記事のコンテンツ・タイプを定義します。可能な値:'プレインテキスト;charset=ISO-8859-15'又は他の有効なcharsetとMIMEタイプ。

「Subject」は記事タイトルを定義します。必須項目。

「Body」は記事内容を定義します。必須項目。

「HistoryType」は、履歴エントリーのタイプを定義します。可能な値：AddNote, ArchiveFlagUpdate, Bounce, CustomerUpdate, EmailAgent,EmailCustomer, EscalationResponseTimeNotifyBefore, EscalationResponseTimeStart,EscalationResponseTimeStop,EscalationSolutionTimeNotifyBefore,Escalat EscalationSolutionTimeStop,EscalationUpdateTimeNotifyBefore,EscalationUpdateTimeStart, EscalationUpdateTimeStop,FollowUp, Forward, Lock, LoopProtection, Merged, Misc, Move, NewTicket, OwnerUpdate,PhoneCallAgent, PhoneCallCustomer,PriorityUpdate,

Remove, ResponsibleUpdate,SendAgentNotification, SendAnswer, SendAutoFollowUp, SendAutoReject, SendAutoReply,SendCustomerNotification, ServiceUpdate, SetPendingTime, SLAUpdate, StateUpdate,Subscribe, SystemRequest, TicketDynamicFieldUpdate, TicketLinkAdd, TicketLinkDelete,TimeAccounting, TypeUpdate, Unlock, Unsubscribe, WebRequestCustomer.

「HistoryComment」は、履歴エントリーの内容を定義します。

上で指定した表記法で'From', 'To', 'Cc' and 'ReplyTo'はEメール・アドレスをとります。

「InReplyTo」および「References」はEメール・メッセージIDをとります。

「NoAgentNotify」-1にセットされれば、担当者への電子メール通知は送られません。

「AutoResponseType」は次の値をとることができます:自動フォローアップ、自動拒否、自動削除、自動応答、自動返答/新規チケット。

'ForceNotificationToUserID', 'ExcludeNotificationToUserID','ExcludeMuteNotificationToUserID' は、ユーザーIDのリストをとることができます。そのリストとは、「常に通知」、「非通知」、または「通知されるが実際には通知メールを送らない、です。

2.4.5.6.3. TicketCustomerSet (チケット・顧客・セット)

顧客にプロセス・チケットをセットします。例：

```
$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」は、設定された推移アクション名前を指定します。

No あるいは、CustomerIDは、顧客の顧客IDをセットします。

ユーザまたはCustomerUserIDは、顧客のユーザー名をセットします。

2.4.5.6.4. TicketLockSet (チケット・ロック・セット)

プロセス・チケットのロックを変更します。例:

```
$Self->{'Process::TransitionAction'} = {
    'TA1' => {
        Name   => 'Set Lock to lock',
        Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketLockSet',
        Config => {
            Lock   => 'lock',
            # or
            LockID => 2,
        },
    },
};
```

「Name」は、設定された推移アクション名前を指定します。

「Lock」は、プロセス・チケットの新しいロックを定義します。

「LockID」は、新しいロックの内部IDを定義します。

2.4.5.6.5. TicketOwnerSet（チケット・所有者・セット）

プロセス・チケットの所有者を変更します。

```
$Self->{'Process::TransitionAction'} = {
    'TA1' => {
        Name   => 'Owner Set root@localhost',
        Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketOwnerSet',
        Config => {
            Owner => 'root@localhost',
            # or
            OwnerID => 1,
        },
    },
};
```

「Name」は、設定された推移アクション名前を指定します。

「Owner」は、新しい所有者のログイン名を指定します。

「OwnerID」は、新しい所有者の内部IDを指定します。

2.4.5.6.6. TicketQueueSet（チケット・キュー・セット）

ターゲット・キューにチケットを移動します。

```
$Self->{'Process::TransitionAction'} = {
    'TA1' => {
        Name   => 'Queue Move Raw',
        Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketQueueSet',
        Config => {
            Queue => 'Raw',
            # or
            # QueueID => '2',
        },
    },
};
```

「Name」は、設定された推移アクション名前を指定します。

「Queue」は、対象のキューの名前を指定します。

「QueueID」は、対象のキューの内部IDを指定します。

2.4.5.6.7. TicketResponsibleSet（チケット・責任者・セット）

プロセス・チケットの責任者を変更します。

```
$Self->{'Process::TransitionAction'} = {
    'TA1' => {
        Name   => 'Responsible Set root@localhost',
        Module =>
 'Kernel::System::ProcessManagement::TransitionAction::TicketResponsibleSet',
        Config => {
            Responsible => 'root@localhost',
            # or
            ResponsibleID => 1,
        },
    },
};
```

```
```

「Name」は、設定された推移アクション名前を指定します。

「Responsible」は、新しい責任者のログイン名を指定します。

「ResponsibleID」は新しい責任者の内部IDを指定します。

2.4.5.6.8. TicketServiceSet（チケット・サービス・セット）

プロセス・チケットにサービスを割り当てます。チケットは顧客を持つことを要求します。そしてそのサービスは、顧客に割り当てなければいけません。例:

```
$Self->{'Process::TransitionAction'} = {
    'TA1' => {
        Name   => 'Set MyService service',
        Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketServiceSet',
        Config => {
            Service   => 'MyService',
            # or
            ServiceID => 123,
        },
    },
};
```

「Name」は、設定された推移アクション名前を指定します。

「Service」は、プロセス・チケットの新しいサービスを定義します。姓名は必要です(例えばGramdFatherService::FatherService::SonService)。

「ServiceID」は、新しいサービスの内部IDを定義します。

2.4.5.6.9. TicketSLASet（チケット・SLA・セット）

プロセス・チケットにサービスレベル合意を割り当てます。チケットはサービスを持つために要求します。また、SLAはそのサービスに割り当てられなければいけません。例:

```
$Self->{'Process::TransitionAction'} = {
    'TA1' => {
        Name   => 'Set MySLA SLA',
        Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketSLASet',
        Config => {
            SLA   => 'MyService',
            # or
            SLAID => 123,
        },
    },
};
```

「Name」は、設定された推移アクション名前を指定します。

「SLA」は、プロセス・チケットの新しいサービスレベル合意を定義します。

「SLAID」は、新しいSLAの内部IDを定義します。

2.4.5.6.10. TicketStateSet（チケット・状態・セット）

プロセス・チケットの状態を変更します。

```
$Self->{'Process::TransitionAction'} = {
    'TA1' => {
        Name   => 'Set State to open',
        Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketStateSet',
        Config => {
```

```
            State   => 'open',
            # or
            StateID => 4,

            PendingTimeDiff => 123,
        },
    },
};
```

「Name」は、設定された推移アクション名前を指定します。

「State」は、プロセス・チケットの新しい状態を定義します。

「StateID」は、新しい状態の内部IDを定義します。

Typeの状態が保留の時のみ使用する「PendingTimeDiff」は、チケットのの保留時間を設定するために（例えば保留時間3600は、推移アクションの実行後、一時間を意味します）、秒単位での時差（推移アクション実行時間に関連して）を定義します。

2.4.5.6.11. TicketTitleSet（チケット・タイトル・セット）

プロセス・チケットのチケット・タイトルをセットします。例:

```
$Self->{'Process::TransitionAction'} = {
    'TA1' => {
        Name   => 'Set Ticket Title to Ticket-title',
        Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketTitleSet',
        Config => {
            Title => 'Ticket-title',
        },
    },
};
```

「Name」は、設定された推移アクション名前を指定します。

「Title」は、チケットの新しいタイトルを指定します。

2.4.5.6.12. TicketTypeSet（チケット・タイプ・セット）

チケット・タイプのプロセス・チケットをセットします。例:

```
$Self->{'Process::TransitionAction'} = {
    'TA1' => {
        Name   => 'Set Ticket Type to default',
        Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketTypeSet',
        Config => {
            Type     => 'default',
            # or
            # TypeID => '1',
        },
    },
};
```

「Name」は、設定された推移アクション名前を指定します。

「Type」は、チケット・タイプの名前を指定します。

「TypeID」は、チケット・タイプの内部IDを指定します。

2.4.6. アクセス・コントロール・リスト(ACL)

ACLの支援によって、プロセス・チケット中の選択可能な値を制限することができます。チケットACLシンタックスの詳細記述については、さらにACLリファレンスを参照ください。

2.4.6.1. ACL構成

ACLは単にKernel/Config.pmに定義することができます。例:

```
$Self->{TicketAcl}->{'001-ACL-ProcessProperties'} = {
    Properties => {
        Process => {
            ProcessEntityID        => ['P1'],
            ActivityEntityID       => ['A1'],
            ActivityDialogEntityID => ['AD1'],
        }
    },
    Possible => {
        ActivityDialog => ['AD1', 'AD3'],
    },
    PossibleNot => {
        ActivityDialog => ['AD3'],
    },
};
```

2.4.6.2. 001-ACL-ProcessProperties

ACL規則の名前。ACL規則一般についてのさらに詳しい情報については、ACLマニュアルを調べてください:

2.4.6.3. プロセス

これはACLが適用されなければいけないかどうかをチェックするために使用されるセクションです。それが指定された値を持っている場合、規則が適用されます。次の値は使用することができます:

2.4.6.3.1. ProcessEntityID

プロセスのID、プロセス。チケットがこのプロセスに割り当てられる場合マッチします。

2.4.6.3.2. ActivityEntityID

プロセス・チケットが現在割り当てられるアクティビティのID。

2.4.6.3.3. ActivityDialogEntityID

プロセス・チケットには現在開いているアクティビティ・ダイアログのID。

2.4.6.4. 「可能」(Possible)/「不可能」(PossibleNot) アクティビティ・ダイアログ

ここでは、アクティビティ・ダイアログIDのリストを指定することができます。このリストは、チケット・ズーム・マスク中のユーザに提示される可能なアクティビティ・ダイアログを制限するでしょう。

「Possible（可能）」は、許可されるアクティビティ・ダイアログをリストします。上記のセッティングは、設定されたアクティビティ・ダイアログのリストの「AD1」および「AD3」のみを許可するでしょう。

「PossibleNot（不可能）」は、許可されないアクティビティ・ダイアログをリストします。上記の例において、セッティングは設定されたアクティビティ・ダイアログのリストから「AD3」を取り除くでしょう。

「可能」と「不可能」の両方が指定されれば、設定されたアクティビティ・ダイアログのリストはまず「可能」でフィルタされます。サンプルではAD1とAD3のみそのまま残してあります。つぎに「不可能」が適用されます。「AD3」はフィルタで除外され、それゆえに「AD1」のみが残り、ユーザが使える可能なアクティビティ・ダイアログとして表示されます。

複数のACL規則がマッチすればしたら、すべてのマッチング規則の共通部分は可能なアクティビティ・ダイアログを決定するために計算されます。例:

設定されたアクティビティ・ダイアログ:「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'],
    },
};
```

プロセス・チケットが状態「new」を持ち、それが「未対応」キューにあり優先度「3　普通」を持つならば、その後、すべてのACL規則がマッチします。

第一のルールはアクティビティ・ダイアログを 'AD1', 'AD2', 'AD3', 'AD4', 'AD5', 'AD6', 'AD7'から'AD1', 'AD2', 'AD3', 'AD6', 'AD7' に減らし、'AD4' and 'AD5'を禁止します。

第二のルールはさらに残っているアクティビティ・ダイアログを減らし、ここでの例の'AD2','AD3', 'AD7'が残ります。。

3番目の規則は、さらに「不可能」によってリストを縮小するでしょう。「AD3」はリストから取り除かれます。それがリストの第一位に載っていなかったので、「AD4」は削除されません。最後に、「AD2」および「AD7」は、ユーザが利用することができる可能なアクティビティ・ダイアログとして残ります。

「新規プロセス・チケット」画面で表示可能なプロセスを限定することは可能です。その機能は、ACLは単にユーザに基づくことができるという唯一の例外をもってActivitiyDialogs（アクティビティ・ダイアログ）を限定することに似ています。

以下のサンプルを御覧ください。

```
$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. 自分のテーマの作成

OTRSウェブ・フロント・エンドの中で好きなレイアウトを使用するように自分のテーマを作成することができます。自分のテーマを作成するために、必要に応じた出力テンプレートをカスタマイズできます。

出力テンプレートのシンタックスおよび構造についてはディベロッパー・マニュアルhttp://otrs.github.io/docでより多くの情報をご覧いただけます。（特にテンプレートについての章の中で）

例として、「Company」と呼ばれる新しいテーマを作成する次のステップを行なってください:

1. Kernel/Output/HTML/Company と呼ばれるディレクトリーを作成して Kernel/Output/HTML/Standard から 新規フォルダへあなたが変えたいと思う全てのファイルをコピーします。

   ### 重要

   実際に変更するファイルを上書きコピーしてください。OTRSは、自動的にスタンダード・テーマから見当たらないファイルを得るでしょう。これは後半の段階でアップグレードすることをより簡単にするでしょう。

2. Kernel/Output/HTML/Company にファイルをカスタマイズします。お好きなようにレイアウトを変更します。

3. 新規のテーマを有効にするにはSysConfig内のFrontend::Themesでテーマを有効にする必要があります

新しいテーマは使用可能に違いありません。個人選択（選好）ページによってそれを選択することができます。

> **警告**
>
> アップデート後にこれらの変更は失われるので、OTRSで送られたテーマ・ファイルを変更しないでください。上に記述されたステップを行なうことによりのみ自分のテーマを作成してください。

4. OTRSフロント・エンドのローカライゼーション

OTRSはそのウェブ・インターフェースのために多重言語サポートを提供します。

OTRSフレームワーク用ローカライゼーションのための手続き、新しい言語変換を作成するために続くステップ、と同様に、翻訳カスタマイズのための手続き、開発者マニュアルhttp://otrs.github.io/doc の "Language Translations（言語変換）"章で見つけることができます。

第6章 パフォーマンス・チューニング

以下はOTRS導入のための強化テクニックのパフォーマンスのリストです。構成、コーディング、メモリー使用、その他を含みます。

1. OTRS

OTRSのパフォーマンスを改善するいくつかのオプションがあります。

1.1. TicketIndexModule

チケット・インデックス用の2つのバックエンド・モジュールがあります:

- Kernel::System::Ticket::IndexAccelerator::RuntimeDB (default)を使うことで、チケット・テーブルから取り急ぎ見えるいくつかのキューを生成します。システム内で約60,000の対応中チケットを持つまではパフォーマンスに関するトラブルを抱えないでしょう。
- あなたが80,000以上の対応中チケットを持っている場合、Kernel::System::Ticket::IndexAccelerator::StaticDB(最も強力なモジュール)が使用されるべきです。それは余分なticket_indexテーブルを使用します。それはビューのように働きます。バックエンドを切り替えた後に最初のインデック作成を生成するためにbin/otrs.RebuildTicketIndex.plを使用してください。

SysConfig経由でIndexAcceleratorを変更することができます。

1.2. TicketStorageModule (チケット・格納・モジュール)

チケット・記事記憶装置用の2つの異なるバックエンド・モジュールがあります:

- 付属などを格納するためにデータベースの中でKernel::System::Ticket::ArticleStorageDB(デフォルト)を形成してください。ノート:大きな組織でそれを使用しないでください。

  賛成意見:ウェブサーバー・ユーザが「otrs」ユーザでない場合は、ファイル許可問題を回避するためにこのモジュールを使用してください。

  反対意見:あなたのデータベースに添付ファイルをを格納することは望ましくありません。あなたのデータベースは大きなオブジェクトを格納できることに気を配って下さい。例えば8MBのオブジェクト(デフォルトは2MBです)を格納するために「set-variable = max_allowed_packet=8M」でMySQLを設定します。

- ローカルのファイルシステム上に添付ファイルなどを格納するためにKernel::System::Ticket::ArticleStorageFSを設定してください。

  賛成意見: 速いですね!

  反対意見:ウェブサーバ・ユーザは「otrs」ユーザであるべきです。さらに、多数のフロントエンドのサーバーを持つなら、、ファイルシステムがサーバー間で共有されることを確かめるべきです。NFS共有か可能ならSANかそれに類似のソリューションに置いて下さい。

取り急ぎ1つのバックエンドから別のバックエンドまで切り替えることができます。SysConfigの中のバックエンドを切り替えて、次に、コマンドライン・ユーティリティotrs.ArticleStorageSwitch.plを実行することができます。データベースからファイルシステムに記事をputするためです。ソースとバックエンドの目的地を指定するために「*-s*」と「*-d*」オプションを使うことができます。全プロセスを実行するには相当な時間がかかることにご注

意ください。それは保持している記事数と可能なCPUパワー及びあるいはネットワーク能力に依存します。

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

スクリプト:記憶バックエンドをデータベースからファイルシステムまで切り替えること。

1.3. チケットのアーカイブに保管

監査に耐えうるシステムとしてOTRSを使うならば、完了したチケットを削除することはよい考えとはいえません。そのため我々はチケットをアーカイブすることができる機能を実装しました。

監査に耐えうるシステムとしてOTRSを使うならば、完了したチケットを削除することはよい考えとはいえません。そのため我々はチケットをアーカイブすることができる機能を実装しました。ある基準とマッチするチケットは「アーカイブ済」としてマークすることができます。これらのチケットは、通常チケット検索、あるいは一般担当者ジョブを実行してもアクセスできません。OTRSを使う時は、「最新の」チケットのみを考慮に入れればよいのでシステムそれ自身は、もはや巨大な量のチケットを取り扱う必要はありません。これにより巨大なシステムでの巨大なパフォーマンスの利得が生まれることとなります。

アーカイブ機能を使用するには次のステップに従います:

1. SysConfigのアーカイブシステムを活性化します。

   管理者ページで、SysConfigに行って、グループ・チケットを選択してください。Core::Ticketでは、デフォルトで「no」にセットされるオプションTicket::ArchiveSystemを見つけます。この設定を「yes」に変更し変更を保存します。

2. GenericAgentジョブを定義する

   管理者ページでGenericAgentを選択し新規ジョブをそこへ追加します。

   a. ジョブ設定

      アーカイブに保管するジョブの名前を与え、この仕事を予定する適切なオプションを選択してください。

   b. チケット・フィルタ

      チケット・フィルタは選択された基準と一致するチケット検索です。完了した状態（数か月前に完了したた）のチケットのみをアーカイブするのは良い考えです。

   c. チケット・アクション

      このセクションでは、「選択したチケットをアーカイブ」「チケットをアーカイブ」とラベルしたフィールドをセットしてください。

   d. 仕事を保存する

      ページの終わりに、仕事を保存するオプションを見つけるでしょう。

   e. 影響を受けたチケット

      システムは、一般担当者のジョブを実行する時にアーカイブされる全てのチケットを表示します。

3. チケット検索

   チケットを検索する場合、システムはデフォルトでアーカイブに保管されないチケットを探索します。さらにアーカイブに保管されたチケットを捜したい場合は、検索基準を定義する時に単に「archive search」を加えてください。

2. データベース

DB発行は使用されているデータベースによって変わります。データベース管理者とデータベースかチェックのためのドキュメンテーションを研究してください。

2.1. MySQL

MySQLテーブル・タイプMyISAM(デフォルトです)を使用し、テーブルの大部分を削除したならば、あるいは可変長列(VARCHAR、BLOBあるいはTEXTカラムがあるテーブル)でテーブルに多くの変更を加えていれば、「optimize（最適化）」コマンドでデータ・ファイル(テーブル)をフラグメンテーション解消しなければいけません。

mysqldデーモンがCPU時間の多くを必要とする場合、テーブルの最適化を試みるべきです。- チケット、ticket_historyおよび記事(下記のスクリプトを参照)。

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

スクリプト:データ・ベース・テーブルの最適化。

2.2. PostgreSQL

PostgreSQLは、PostgreSQLデータディレクトリのpostgresql.confファイルの修正により最適化されます。これをする方法についてのアドバイスについては、次の記事を参考文献としてください:

- http://www.revsys.com/writings/postgresql-performance.html
- http://varlena.com/GeneralBits/Tidbits/perf.html
- http://varlena.com/GeneralBits/Tidbits/annotated_conf_e.html

パフォーマンスが不十分な場合、PostgreSQLパフォーマンス・メーリング・リスト(http://www.postgresql.org/community/lists/)に参加しそこに質問をすることを提案します。PostgreSQLリスト上の人々は非常にフレンドリーで、おそらく助けてくれることでしょう。

3. ウェブサーバ

mod_perl 2.0(http://perl.apache.org/)を使用するべきです。それは純粋なcgiよりはるかに速い(~ * 100)。しかし、それはより多くのRAMを必要とします。

3.1. 事前に確立しているデータベース接続

ウェブサーバのスタートアップに先んじてあらかじめデータベース接続を確立することができます。時間の節約になります(README.webserverを参照)。

3.2. プレインストールされたモジュール - startup.pl

スタートアップ・スクリプト・スクリプトscripts/apache2-perl-startup.plを使用して下さい。それはmod_perlウェブサーバ上の事前ロードされ、コンパイルされたPerlをより早くします。しかもより小さいメモリで。(README.webserverを参照)。

3.3. ディスク上で更新された時パール・モジュールをリロードします。

Apache::Reloadはデフォルトでscripts/apache2-httpd.include.conf.の中で使用されます。それを不能にすれば8%のスピードアップとなります。しかし、もしOTRSパッケージマネージャから

なんらかのモジュールを、あるいは SysConfig 中または Kernel/Config.pm.中でなんらかの値を、インストールしているならば、忘れてはいけないことは
ウェブサーバのリスタートです。重要:このことは、ウェブ・インターフェース経由でOTRSパッケージ・マネージャーを使用することができないことを意味します。コマンドラインから変数の -bin/otrs.PackageManager.plをご利用下さい。

3.4. 正しい戦略の選択

より大きな装置(例えば40人以上担当者または1,000枚/day以上の新しいチケット)を持っているなら、mod_perlユーザ・ガイド(http://perl.apache.org/docs/2.0/user/index.html)の性能についての章を読むと良いでしょう。

3.5. mod_gzip/mod_deflate

帯域幅が小さい場合は、Apache2用のmod_deflateをご利用ください。45000のHTMLページを持っているなら、mod_gzip/mod_deflate
は約7000にそれを圧縮します。欠点は、これがサーバー側のロードを増加させるということです。

付録A 追加の資料

私達はOTRSに関する正に最新情報でサポートしようとします。さらにフィードバックを提供する機会も与えます。

1. OTRSグループのウェブサイト

*http://www.otrs.com*で、OTRSグループ(OTRSを支える集団)のウェブサイトを見つけることができます。それは、白書、リリース・ノート、事例研究などのような多くの製品関連情報を含んでいます。

2. メーリングリスト

テーブルA-1はさまざまなコミュニティのメーリングリストを表示します。

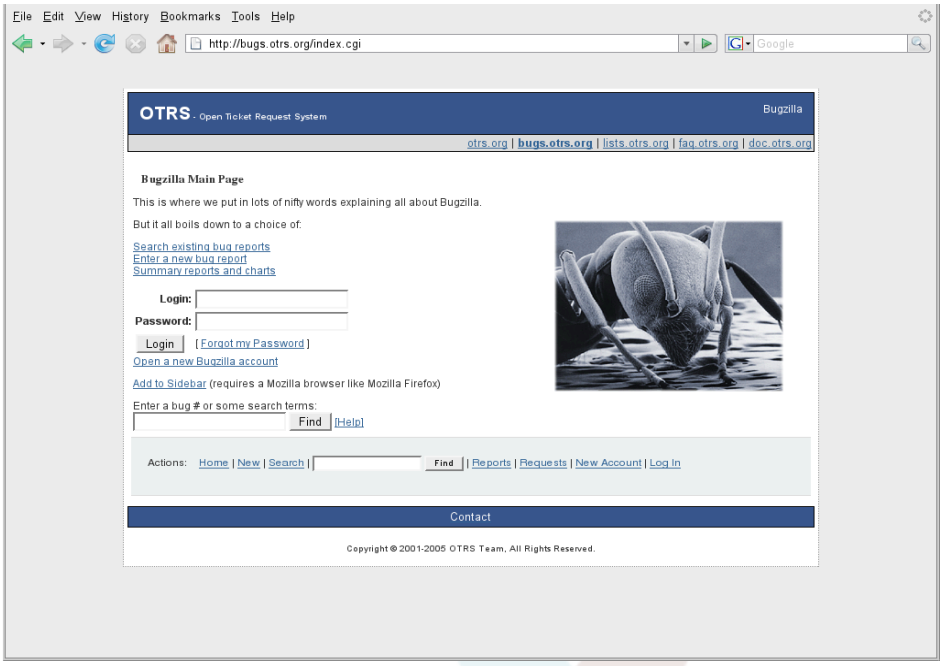
表A.1 メーリングリスト

Name	摘要	ホームページ
announce@otrs.org	新しいOTRSリリースおよびセキュリティ問題に関するアナウンスのためリスト。英語で、転送量は少量です。	http://lists.otrs.org/cgi-bin/listinfo/announce
otrs@otrs.org	全ての関連質問および製品サポートについてのリスト。英語で、流量は中/多量です。	http://lists.otrs.org/cgi-bin/listinfo/otrs
otrs-de@otrs.org	全ての関連質問および製品サポートについてのリスト。ドイツ語で、流量は中/多量です。	http://lists.otrs.org/cgi-bin/listinfo/otrs-de
dev@otrs.org	OTRS開発者がさまざまなデザインや実装の問題を議論するリスト。英語で、流量は中度です。	http://lists.otrs.org/cgi-bin/listinfo/dev
i18n@otrs.org	国際化および現地化に関する質問に関して。OTRSプロジェクトの翻訳者になりたいか、国際的環境でアプリケーションに関してどんな問題でもあれば、接続するのはここです。英語で、流量は少量です。	http://lists.otrs.org/cgi-bin/listinfo/i18n

これらのメーリングリストのいずれかを購読するには以下のリンクを訪問して下さい。http://lists.otrs.org/。

3. バグ・トラッキング

バグを送信するためにhttp://bugs.otrs.org/を訪ねて下さい(下記の図を参照)。バグと構成問題間の違いについてノートをとってください。構成問題は、システムを設定する時遭遇する問題、あるいはOTRSの使用に関する一般的な質問です。バグ・レポートは、OTRSのソース・コード自体の問題あるいはOTRSの改善をファイルするためのみに使用されるべきです。全てのバグ・レポートおよび改善要求を、バグ・トラッカー非常に歓迎します。



図：バグ・トラッカー

構成問題については、*OTRS.comから利用可能な商用サポート*、あるいは公式メーリング・リストのどちらかを利用すべきです。

みなさんのバグ・レポートによって製品が改善されます。みなさんのインプットに感謝します！

4. 商用サポート

サービス（サポート、コンサルティング、開発およびトレーニング）については、OTRSの支援団体、OTRS　AGと連絡をとることができます。私たちのオフィスはドイツ、アメリカ、メキシコ、オランダおよび他の国々に位置します。連絡先情報のために私たちのウェブサイトを訪れてください:http://www.otrs.com/en/corporate-navigation/contact/

付録B Configuration Options Reference

1. DynamicFields

1.1. DynamicFields::Driver::Registration

1.1.1. DynamicFields::Driver###Text

Description:	DynamicField backend registration.
Group:	DynamicFields
SubGroup:	DynamicFields::Driver::Registration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'DynamicFields::Driver'}->{'Text'} = {`
	`'ConfigDialog' => 'AdminDynamicFieldText',`
	`'DisplayName' => 'Text',`
	`'Module' => 'Kernel::System::DynamicField::Driver::Text'`
	`};`

1.1.2. DynamicFields::Driver###TextArea

Description:	DynamicField backend registration.
Group:	DynamicFields
SubGroup:	DynamicFields::Driver::Registration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'DynamicFields::Driver'}->{'TextArea'} = {`
	`'ConfigDialog' => 'AdminDynamicFieldText',`
	`'DisplayName' => 'Textarea',`
	`'Module' => 'Kernel::System::DynamicField::Driver::TextArea'`
	`};`

1.1.3. DynamicFields::Driver###Checkbox

Description:	DynamicField backend registration.
Group:	DynamicFields
SubGroup:	DynamicFields::Driver::Registration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'DynamicFields::Driver'}->{'Checkbox'} = {`
	`'ConfigDialog' => 'AdminDynamicFieldCheckbox',`
	`'DisplayName' => 'Checkbox',`
	`'Module' => 'Kernel::System::DynamicField::Driver::Checkbox'`
	`};`

1.1.4. DynamicFields::Driver###Dropdown

Description:	DynamicField backend registration.
Group:	DynamicFields

SubGroup:	DynamicFields::Driver::Registration
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'DynamicFields::Driver'}->{'Dropdown'} = {
	'ConfigDialog' => 'AdminDynamicFieldDropdown',
	'DisplayName' => 'Dropdown',
	'Module' => 'Kernel::System::DynamicField::Driver::Dropdown'
	};</pre>

1.1.5. DynamicFields::Driver###DateTime

Description:	DynamicField backend registration.
Group:	DynamicFields
SubGroup:	DynamicFields::Driver::Registration
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'DynamicFields::Driver'}->{'DateTime'} = {
	'ConfigDialog' => 'AdminDynamicFieldDateTime',
	'DisplayName' => 'Date / Time',
	'Module' => 'Kernel::System::DynamicField::Driver::DateTime'
	};</pre>

1.1.6. DynamicFields::Driver###Date

Description:	DynamicField backend registration.
Group:	DynamicFields
SubGroup:	DynamicFields::Driver::Registration
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'DynamicFields::Driver'}->{'Date'} = {
	'ConfigDialog' => 'AdminDynamicFieldDateTime',
	'DisplayName' => 'Date',
	'Module' => 'Kernel::System::DynamicField::Driver::Date'
	};</pre>

1.1.7. DynamicFields::Driver###Multiselect

Description:	DynamicField backend registration.
Group:	DynamicFields
SubGroup:	DynamicFields::Driver::Registration
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'DynamicFields::Driver'}->{'Multiselect'} = {
	'ConfigDialog' => 'AdminDynamicFieldMultiselect',
	'DisplayName' => 'Multiselect',
	'ItemSeparator' => ', ',
	'Module' => 'Kernel::System::DynamicField::Driver::Multiselect'
	};</pre>

1.2. DynamicFields::ObjectType::Registration

1.2.1. DynamicFields::ObjectType###Ticket

Description:	DynamicField object registration.

Group:	DynamicFields
SubGroup:	DynamicFields::ObjectType::Registration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'DynamicFields::ObjectType'}->{'Ticket'} = {`
	`'DisplayName' => 'Ticket',`
	`'Module' => 'Kernel::System::DynamicField::ObjectType::Ticket'`
	`};`

1.2.2. DynamicFields::ObjectType###Article

Description:	DynamicField object registration.
Group:	DynamicFields
SubGroup:	DynamicFields::ObjectType::Registration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'DynamicFields::ObjectType'}->{'Article'} = {`
	`'DisplayName' => 'Article',`
	`'Module' => 'Kernel::System::DynamicField::ObjectType::Article'`
	`};`

1.3. Frontend::Admin::ModuleRegistration

1.3.1. Frontend::Module###AdminDynamicField

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

Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminDynamicField'} = {`
	`'Description' => 'Admin',`
	`'Group' => [`
	`'admin'`
	`],`
	`'Loader' => {`
	`'CSS' => [`
	`'Core.Agent.Admin.DynamicField.css'`
	`],`
	`'JavaScript' => [`
	`'Core.Agent.Admin.DynamicField.js'`
	`]`
	`},`
	`'NavBarModule' => {`
	`'Block' => 'Ticket',`
	`'Description' => 'Create and manage dynamic fields.',`
	`'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',`
	`'Name' => 'Dynamic Fields',`
	`'Prio' => '1000'`
	`},`
	`'NavBarName' => 'Admin',`
	`'Title' => 'Dynamic Fields GUI'`
	`};`

1.3.2. Frontend::Module###AdminDynamicFieldText

Description:	Frontend module registration for the agent interface.
Group:	DynamicFields

SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AdminDynamicFieldText'} = {
	'Description' => 'Admin',
	'Group' => [
	'admin'
	],
	'Loader' => {
	'JavaScript' => [
	'Core.Agent.Admin.DynamicField.js'
	]
	},
	'Title' => 'Dynamic Fields Text Backend GUI'
	};</pre>

1.3.3. Frontend::Module###AdminDynamicFieldCheckbox

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

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AdminDynamicFieldCheckbox'} = {
	'Description' => 'Admin',
	'Group' => [
	'admin'
	],
	'Loader' => {
	'JavaScript' => [
	'Core.Agent.Admin.DynamicField.js'
	]
	},
	'Title' => 'Dynamic Fields Checkbox Backend GUI'
	};</pre>

1.3.4. Frontend::Module###AdminDynamicFieldDropdown

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

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AdminDynamicFieldDropdown'} = {
	'Description' => 'Admin',
	'Group' => [
	'admin'
	],
	'Loader' => {
	'CSS' => [
	'Core.Agent.Admin.DynamicField.css'
	],
	'JavaScript' => [
	'Core.Agent.Admin.DynamicField.js',
	'Core.Agent.Admin.DynamicFieldDropdown.js'
	]
	},</pre>

```
  'Title' => 'Dynamic Fields Drop-down Backend GUI'
};
```

1.3.5. Frontend::Module###AdminDynamicFieldDateTime

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

```
$Self->{'Frontend::Module'}->{'AdminDynamicFieldDateTime'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.DynamicField.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.DynamicField.js',
      'Core.Agent.Admin.DynamicFieldDateTime.js'
    ]
  },
  'Title' => 'Dynamic Fields Date Time Backend GUI'
};
```

1.3.6. Frontend::Module###AdminDynamicFieldMultiselect

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

```
$Self->{'Frontend::Module'}->{'AdminDynamicFieldMultiselect'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.DynamicField.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.DynamicField.js',
      'Core.Agent.Admin.DynamicFieldMultiselect.js'
    ]
  },
  'Title' => 'Dynamic Fields Multiselect Backend GUI'
};
```

1.4. Frontend::Agent::Preferences

1.4.1. PreferencesGroups###DynamicField

Description:	Defines the config parameters of this item, to be shown in the preferences view.

Group:	DynamicFields
SubGroup:	Frontend::Agent::Preferences
Valid:	0
Required:	0

Config-Setting:	`$Self->{'PreferencesGroups'}->{'DynamicField'} = {`
	`'Active' => '1',`
	`'Block' => 'Input',`
	`'Column' => 'Other Settings',`
	`'Data' => '$Env{"UserDynamicField_NameX"}',`
	`'Key' => 'Default value for NameX',`
	`'Label' => 'NameX',`
	`'Module' => 'Kernel::Output::HTML::PreferencesGeneric',`
	`'PrefKey' => 'UserDynamicField_NameX',`
	`'Prio' => '7000'`
	`};`

1.4.2. PreferencesGroups###DynamicFieldsOverviewPageShown

Description:	Parameters for the pages (in which the dynamic fields are shown) of the dynamic fields overview.
Group:	DynamicFields
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	`$Self->{'PreferencesGroups'}->{'DynamicFieldsOverviewPageShown'} =`
	`{`
	`'Active' => '0',`
	`'Column' => 'Other Settings',`
	`'Data' => {`
	`'10' => '10',`
	`'15' => '15',`
	`'20' => '20',`
	`'25' => '25',`
	`'30' => '30',`
	`'35' => '35'`
	`},`
	`'DataSelected' => '25',`
	`'Key' => 'Dynamic fields limit per page for Dynamic Fields Overview',`
	`'Label' => 'Dynamic Fields Overview Limit',`
	`'Module' => 'Kernel::Output::HTML::PreferencesGeneric',`
	`'PrefKey' => 'AdminDynamicFieldsOverviewPageShown',`
	`'Prio' => '8000'`
	`};`

2. Framework

2.1. Core

2.1.1. SecureMode

Description:	Disables the web installer (http://yourhost.example.com/otrs/installer.pl), to prevent the system from being hijacked. If set to "No", the system can be reinstalled and the current basic configuration will be used to pre-populate the questions within the installer script. If not active, it also disables the GenericAgent, PackageManager and SQL Box.

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

2.1.2. Frontend::DebugMode

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

2.1.3. ConfigLevel

Description:	Sets the configuration level of the administrator. Depending on the config level, some sysconfig options will be not shown. The config levels are in in ascending order: Expert, Advanced, Beginner. The higher the config level is (e.g. Beginner is the highest), the less likely is it that the user can accidentally configure the system in a way that it is not usable any more.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'ConfigLevel'} =  '100';`

2.1.4. ProductName

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

2.1.5. SystemID

Description:	Defines the system identifier. Every ticket number and http session string contains this ID. This ensures that only tickets which belong to your system will be processed as follow-ups (useful when communicating between two instances of OTRS).
Group:	Framework
SubGroup:	Core

Valid:	1
Required:	1
Config-Setting:	`$Self->{'SystemID'} =  '10';`

2.1.6. FQDN

Description:	Defines the fully qualified domain name of the system. This setting is used as a variable, OTRS_CONFIG_FQDN which is found in all forms of messaging used by the application, to build links to the tickets within your system.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'FQDN'} =  'yourhost.example.com';`

2.1.7. SupportDataCollector::HTTPHostname

Description:	Defines the HTTP hostname for the support data collection with the public module 'PublicSupportDataCollector' (e.g. used from the OTRS Daemon).
Group:	Framework
SubGroup:	Core
Valid:	0
Required:	0
Config-Setting:	`$Self->{'SupportDataCollector::HTTPHostname'} =  '';`

2.1.8. HttpType

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

2.1.9. ScriptAlias

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

	OTRS_CONFIG_ScriptAlias which is found in all forms of messaging used by the application, to build links to the tickets within the system.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'ScriptAlias'} = 'otrs/';`

2.1.10. AdminEmail

Description:	Defines the system administrator's email address. It will be displayed in the error screens of the application.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'AdminEmail'} = 'admin@example.com';`

2.1.11. Organization

Description:	Company name which will be included in outgoing emails as an X-Header.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Organization'} = 'Example Company';`

2.1.12. DefaultLanguage

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

2.1.13. DefaultUsedLanguages

Description:	Defines all the languages that are available to the application. The Key/Content pair links the front-end display name to the appropriate language PM file. The "Key" value should be the base-name of the PM file (i.e. de.pm is the file, then de is the "Key" value). The "Content" value should be the display

	name for the front-end. Specify any own-defined language here (see the developer documentation http://doc.otrs.org/ for more infomation). Please remember to use the HTML equivalents for non-ASCII characters (i.e. for the German oe = o umlaut, it is necessary to use the ö symbol).
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1

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

2.1.14. DefaultTheme

Description:	Defines the default front-end (HTML) theme to be used by the agents and customers. If you like, you can add your own theme. Please refer the administrator manual located at http://doc.otrs.org/.

Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'DefaultTheme'} = 'Standard';`

2.1.15. DefaultTheme::HostBased

Description:	It is possible to configure different themes, for example to distinguish between agents and customers, to be used on a per-domain basis within the application. Using a regular expression (regex), you can configure a Key/Content pair to match a domain. The value in "Key" should match the domain, and the value in "Content" should be a valid theme on your system. Please see the example entries for the proper form of the regex.
Group:	Framework
SubGroup:	Core
Valid:	0
Required:	0
Config-Setting:	`$Self->{'DefaultTheme::HostBased'} = { 'host1\\.example\\.com' => 'SomeTheme1', 'host2\\.example\\.com' => 'SomeTheme2' };`

2.1.16. CheckMXRecord

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

2.1.17. CheckMXRecord::Nameserver

Description:	Defines the address of a dedicated DNS server, if necessary, for the "CheckMXRecord" look-ups.
Group:	Framework
SubGroup:	Core
Valid:	0
Required:	0
Config-Setting:	`$Self->{'CheckMXRecord::Nameserver'} = 'ns.example.com';`

2.1.18. CheckEmailAddresses

Description:	Makes the application check the syntax of email addresses.
Group:	Framework

SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CheckEmailAddresses'} = '1';`

2.1.19. CheckEmailValidAddress

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

2.1.20. CheckEmailInvalidAddress

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

2.1.21. CGILogPrefix

Description:	Specifies the text that should appear in the log file to denote a CGI script entry.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CGILogPrefix'} = 'OTRS-CGI';`

2.1.22. DemoSystem

Description:	Runs the system in "Demo" mode. If set to "Yes", agents can change preferences, such as selection of language and theme via the agent web interface. These changes are only valid for the current session. It will not be possible for agents to change their passwords.
Group:	Framework
SubGroup:	Core

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

2.1.23. SwitchToUser

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

2.1.24. SwitchToCustomer

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

2.1.25. SwitchToCustomer::PermissionGroup

Description:	Specifies the group where the user needs rw permissions so that he can access the "SwitchToCustomer" feature.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SwitchToCustomer::PermissionGroup'} =  'admin';`

2.1.26. NotificationSenderName

Description:	Specifies the name that should be used by the application when sending notifications. The sender name is used to build the complete display name for the notification master (i.e. "OTRS Notification Master" otrs@your.example.com). Notifications are messages such as en::Customer::QueueUpdate or en::Agent::Move.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'NotificationSenderName'} =  'OTRS Notification Master';`

2.1.27. NotificationSenderEmail

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

2.1.28. System::Customer::Permission

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

2.1.29. LanguageDebug

Description:	Debugs the translation set. If this is set to "Yes" all strings (text) without translations are written to STDERR. This can be helpful when you are creating a new translation file. Otherwise, this option should remain set to "No".
Group:	Framework
SubGroup:	Core
Valid:	0
Required:	0
Config-Setting:	`$Self->{'LanguageDebug'} =  '0';`

2.1.30. Secure::DisableBanner

Description:	If enabled, the OTRS version tag will be removed from the Webinterface, the HTTP headers and the X-Headers of outgoing mails.
Group:	Framework
SubGroup:	Core

Valid:	0
Required:	0
Config-Setting:	`$Self->{'Secure::DisableBanner'} = '0';`

2.1.31. StandardTemplate2QueueByCreating

Description:	List of default Standard Templates which are assigned automatically to new Queues upon creation.
Group:	Framework
SubGroup:	Core
Valid:	0
Required:	0
Config-Setting:	`$Self->{'StandardTemplate2QueueByCreating'} = [ '' ];`

2.2. Core::Cache

2.2.1. Cache::Module

Description:	Selects the cache backend to use.
Group:	Framework
SubGroup:	Core::Cache
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Cache::Module'} = 'Kernel::System::Cache::FileStorable';`

2.2.2. Cache::SubdirLevels

Description:	Specify how many sub directory levels to use when creating cache files. This should prevent too many cache files being in one directory.
Group:	Framework
SubGroup:	Core::Cache
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Cache::SubdirLevels'} = '2';`

2.3. Core::CustomerCompany

2.3.1. CustomerCompany::EventModulePost###100-UpdateCustomerUsers

Description:	Event module that updates customer users after an update of the Customer Company.
Group:	Framework
SubGroup:	Core::CustomerCompany
Valid:	1

Required:	0

Config-Setting:	`$Self->{'CustomerCompany::EventModulePost'}->{'100-UpdateCustomerUsers'} = {`
	`'Event' => 'CustomerCompanyUpdate',`
	`'Module' => 'Kernel::System::CustomerCompany::Event::CustomerUserUpdate',`
	`'Transaction' => '0'`
	`};`

2.4. Core::CustomerUser

2.4.1. CustomerUser::EventModulePost###100-UpdateServiceMembership

Description:	Event module that updates customer user service membership if login changes.
Group:	Framework
SubGroup:	Core::CustomerUser
Valid:	1
Required:	0

Config-Setting:	`$Self->{'CustomerUser::EventModulePost'}->{'100-UpdateServiceMembership'} = {`
	`'Event' => 'CustomerUserUpdate',`
	`'Module' => 'Kernel::System::CustomerUser::Event::ServiceMemberUpdate',`
	`'Transaction' => '0'`
	`};`

2.5. Core::LinkObject

2.5.1. LinkObject::ViewMode

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

2.5.2. LinkObject::Type###Normal

Description:	Defines the link type 'Normal'. If the source name and the target name contain the same value, the resulting link is a non-directional one; otherwise, the result is a directional link.
Group:	Framework
SubGroup:	Core::LinkObject
Valid:	1
Required:	1

Config-Setting:	`$Self->{'LinkObject::Type'}->{'Normal'} = {`
	`'SourceName' => 'Normal',`
	`'TargetName' => 'Normal'`
	`};`

2.5.3. LinkObject::Type###ParentChild

Description:	Defines the link type 'ParentChild'. If the source name and the target name contain the same value, the resulting link is a non-directional one; otherwise, the result is a directional link.
Group:	Framework
SubGroup:	Core::LinkObject
Valid:	1
Required:	1

Config-Setting:	`$Self->{'LinkObject::Type'}->{'ParentChild'} = {`
	`  'SourceName' => 'Parent',`
	`  'TargetName' => 'Child'`
	`};`

2.5.4. LinkObject::TypeGroup###0001

Description:	Defines the link type groups. The link types of the same group cancel one another. Example: If ticket A is linked per a 'Normal' link with ticket B, then these tickets could not be additionally linked with link of a 'ParentChild' relationship.
Group:	Framework
SubGroup:	Core::LinkObject
Valid:	1
Required:	0

Config-Setting:	`$Self->{'LinkObject::TypeGroup'}->{'0001'} = [`
	`  'Normal',`
	`  'ParentChild'`
	`];`

2.6. Core::Log

2.6.1. LogModule

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

2.6.2. LogModule::SysLog::Facility

Description:	If "SysLog" was selected for LogModule, a special log facility can be specified.
Group:	Framework
SubGroup:	Core::Log
Valid:	1
Required:	1

Config-Setting:	`$Self->{'LogModule::SysLog::Facility'} = 'user';`

2.6.3. LogModule::SysLog::LogSock

Description:	If "SysLog" was selected for LogModule, a special log sock can be specified (on solaris you may need to use 'stream').
Group:	Framework
SubGroup:	Core::Log
Valid:	1
Required:	1
Config-Setting:	`$Self->{'LogModule::SysLog::LogSock'} = 'unix';`

2.6.4. LogModule::SysLog::Charset

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

2.6.5. LogModule::LogFile

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

2.6.6. LogModule::LogFile::Date

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

2.7. Core::MIME-Viewer

2.7.1. MIME-Viewer###application/excel

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

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

2.7.2. MIME-Viewer###application/msword

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

2.7.3. MIME-Viewer###application/pdf

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

2.7.4. MIME-Viewer###text/xml

Description:	Specifies the path to the converter that allows the view of XML files, in the web interface.
Group:	Framework
SubGroup:	Core::MIME-Viewer
Valid:	0
Required:	0
Config-Setting:	`$Self->{'MIME-Viewer'}->{'text/xml'} = '<OTRS_CONFIG_Home>/scripts/tools/xml2html.pl';`

2.8. Core::MirrorDB

2.8.1. Core::MirrorDB::DSN

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

Required:	0
Config-Setting:	`$Self->{'Core::MirrorDB::DSN'} = 'DBI:mysql:database=mirrordb;host=mirrordbhost';`

2.8.2. Core::MirrorDB::User

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

2.8.3. Core::MirrorDB::Password

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

2.9. Core::PDF

2.9.1. PDF

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

2.9.2. PDF::LogoFile

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

2.9.3. PDF::PageSize

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

2.9.4. PDF::MaxPages

Description:	Defines the maximum number of pages per PDF file.
Group:	Framework
SubGroup:	Core::PDF
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PDF::MaxPages'} =  '100';`

2.9.5. PDF::TTFontFile###Proportional

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

2.9.6. PDF::TTFontFile###ProportionalBold

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

2.9.7. PDF::TTFontFile###ProportionalItalic

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

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

2.9.8. PDF::TTFontFile###ProportionalBoldItalic

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

2.9.9. PDF::TTFontFile###Monospaced

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

2.9.10. PDF::TTFontFile###MonospacedBold

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

2.9.11. PDF::TTFontFile###MonospacedItalic

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

2.9.12. PDF::TTFontFile###MonospacedBoldItalic

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

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

2.10. Core::Package

2.10.1. Package::FileUpload

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

2.10.2. Package::RepositoryRoot

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

2.10.3. Package::RepositoryList

Description:	Defines the list of online repositories. Another installations can be used as repository, for example: Key="http://example.com/otrs/public.pl?Action=PublicRepository;File=" and Content="Some Name".
Group:	Framework
SubGroup:	Core::Package
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Package::RepositoryList'} = { 'ftp://ftp.example.com/pub/otrs/misc/packages/' => '[Example] ftp://ftp.example.com/' };`

2.10.4. Package::RepositoryAccessRegExp

Description:	Defines the IP regular expression for accessing the local repository. You need to enable this to have access to your local repository and the package::RepositoryList is required on the remote host.

Group:	Framework
SubGroup:	Core::Package
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Package::RepositoryAccessRegExp'} = '127\\.0\\.0\\.1';`

2.10.5. Package::Timeout

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

2.10.6. Package::Proxy

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

2.10.7. Package::ShowFeatureAddons

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

2.10.8. Package::EventModule###99-SupportDataSend

Description:	Package event module file a scheduler task for update registration.
Group:	Framework
SubGroup:	Core::Package
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Package::EventModule'}->{'99-SupportDataSend'} = { 'Event' => '(PackageInstall|PackageReinstall|PackageUpgrade|PackageUninstall)', 'Module' => 'Kernel::System::Package::Event::SupportDataSend', 'Transaction' => '1'`

	};

2.11. Core::PerformanceLog

2.11.1. PerformanceLog

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

2.11.2. PerformanceLog::File

Description:	Specifies the path of the file for the performance log.
Group:	Framework
SubGroup:	Core::PerformanceLog
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PerformanceLog::File'} = '<OTRS_CONFIG_Home>/var/log/Performance.log';`

2.11.3. PerformanceLog::FileMax

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

2.12. Core::ReferenceData

2.12.1. ReferenceData::OwnCountryList

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

	`};`

2.13. Core::SOAP

2.13.1. SOAP::User

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

2.13.2. SOAP::Password

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

2.13.3. SOAP::Keep-Alive

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

2.14. Core::Sendmail

2.14.1. SendmailModule

Description:	Defines the module to send emails. "Sendmail" directly uses the sendmail binary of your operating system. Any of the "SMTP" mechanisms use a specified (external) mailserver. "DoNotSendEmail" doesn't send emails and it is useful for test systems.
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SendmailModule'} =  'Kernel::System::Email::Sendmail';`

2.14.2. SendmailModule::CMD

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

2.14.3. SendmailModule::Host

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

2.14.4. SendmailModule::Port

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

2.14.5. SendmailModule::AuthUser

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

2.14.6. SendmailModule::AuthPassword

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

Group:	Framework
SubGroup:	Core::Sendmail
Valid:	0
Required:	0
Config-Setting:	`$Self->{'SendmailModule::AuthPassword'} = 'MailserverPassword';`

2.14.7. SendmailBcc

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

2.14.8. SendmailEnvelopeFrom

Description:	If set, this address is used as envelope sender in outgoing messages (not notifications - see below). If no address is specified, the envelope sender is equal to queue e-mail address.
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	0
Required:	0
Config-Setting:	`$Self->{'SendmailEnvelopeFrom'} = '';`

2.14.9. SendmailNotificationEnvelopeFrom

Description:	If set, this address is used as envelope sender header in outgoing notifications. If no address is specified, the envelope sender header is empty.
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	0
Required:	0
Config-Setting:	`$Self->{'SendmailNotificationEnvelopeFrom'} = '';`

2.14.10. SendmailEncodingForce

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

2.15. Core::Session

2.15.1. SessionModule

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

2.15.2. SessionName

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

2.15.3. CustomerPanelSessionName

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

2.15.4. SessionCheckRemoteIP

Description:	Turns on the remote ip address check. It should be set to "No" if the application is used, for example, via a proxy farm or a dialup connection, because the remote ip address is mostly different for the requests.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SessionCheckRemoteIP'} =  '1';`

2.15.5. SessionDeleteIfNotRemoteID

Description:	Deletes a session if the session id is used with an invalid remote IP address.
Group:	Framework

SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SessionDeleteIfNotRemoteID'} = '1';`

2.15.6. SessionMaxTime

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

2.15.7. SessionMaxIdleTime

Description:	Sets the inactivity time (in seconds) to pass before a session is killed and a user is loged out.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SessionMaxIdleTime'} = '21600';`

2.15.8. SessionActiveTime

Description:	Sets the time (in seconds) a user is marked as active.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SessionActiveTime'} = '600';`

2.15.9. SessionDeleteIfTimeToOld

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

2.15.10. SessionUseCookie

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

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

2.15.11. SessionUseCookieAfterBrowserClose

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

2.15.12. SessionCSRFProtection

Description:	Protection against CSRF (Cross Site Request Forgery) exploits (for more info see http://en.wikipedia.org/wiki/Cross-site_request_forgery).
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SessionCSRFProtection'} = '1';`

2.15.13. AgentSessionLimit

Description:	Sets the maximum number of active agents within the timespan defined in SessionActiveTime.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	0
Config-Setting:	`$Self->{'AgentSessionLimit'} = '100';`

2.15.14. CustomerSessionLimit

Description:	Sets the maximum number of active customers within the timespan defined in SessionActiveTime.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerSessionLimit'} = '100';`

2.15.15. SessionDir

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

2.15.16. SessionTable

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

2.16. Core::SpellChecker

2.16.1. SpellChecker

Description:	Enables spell checker support.
Group:	Framework
SubGroup:	Core::SpellChecker
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SpellChecker'} =  '0';`

2.16.2. SpellCheckerBin

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

2.16.3. SpellCheckerDictDefault

Description:	Defines the default spell checker dictionary.
Group:	Framework
SubGroup:	Core::SpellChecker

Valid:	1
Required:	1
Config-Setting:	`$Self->{'SpellCheckerDictDefault'} = 'english';`

2.16.4. SpellCheckerIgnore

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

2.17. Core::Stats

2.17.1. Stats::StatsHook

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

2.17.2. Stats::StatsStartNumber

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

2.17.3. Stats::MaxXaxisAttributes

Description:	Defines the default maximum number of X-axis attributes for the time scale.
Group:	Framework
SubGroup:	Core::Stats
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Stats::MaxXaxisAttributes'} =  '1000';`

2.18. Core::Stats::Graph

2.18.1. Stats::Graph::t_margin

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

2.18.2. Stats::Graph::l_margin

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

2.18.3. Stats::Graph::b_margin

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

2.18.4. Stats::Graph::r_margin

Description:	Specifies the right margin of the chart.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::r_margin'} =  '20';`

2.18.5. Stats::Graph::bgclr

Description:	Specifies the background color of the picture.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Stats::Graph::bgclr'} = 'white';`

2.18.6. Stats::Graph::transparent

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

2.18.7. Stats::Graph::fgclr

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

2.18.8. Stats::Graph::boxclr

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

2.18.9. Stats::Graph::accentclr

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

2.18.10. Stats::Graph::legendclr

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

2.18.11. Stats::Graph::textclr

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

2.18.12. Stats::Graph::dclrs

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

2.18.13. Stats::Graph::TitleFont

Description:	Defines the title font in graphs (place custom fonts in var/fonts).
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::TitleFont'} = 'DejaVuSans-Bold.ttf';`

2.18.14. Stats::Graph::LegendFont

Description:	Defines the legend font in graphs (place custom fonts in var/fonts).
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::LegendFont'} = 'DejaVuSans.ttf';`

2.18.15. Stats::Graph::line_width

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

2.18.16. Stats::Graph::legend_placement

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

2.18.17. Stats::Graph::legend_spacing

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

2.18.18. Stats::Graph::legend_marker_width

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

2.18.19. Stats::Graph::legend_marker_height

Description:	Defines the height of the legend.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Stats::Graph::legend_marker_height'} =  '8';`

2.19. Core::Time

2.19.1. TimeInputFormat

Description:	Defines the date input format used in forms (option or input fields).
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeInputFormat'} =  'Option';`

2.19.2. TimeShowAlwaysLong

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

2.19.3. TimeZone

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

2.19.4. TimeZoneUser

Description:	Sets the user time zone per user (required a system with UTC as system time and UTC under TimeZone). Otherwise this is a diff time to the local time.
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	0
Config-Setting:	`$Self->{'TimeZoneUser'} =  '0';`

2.19.5. TimeZoneUserBrowserAutoOffset

Description:	Sets the user time zone per user based on java script / browser time zone offset feature at login time.

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

2.19.6. CalendarWeekDayStart

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

2.19.7. TimeVacationDays

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

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

2.19.8. TimeVacationDaysOneTime

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

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

2.19.9. TimeWorkingHours

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

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

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

2.19.10. TimeShowCompleteDescription

Description:	Shows time use complete description (days, hours, minutes), if set to "Yes"; or just first letter (d, h, m), if set to "No".
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeShowCompleteDescription'} =  '0';`

2.20. Core::Time::Calendar1

2.20.1. TimeZone::Calendar1Name

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

2.20.2. TimeZone::Calendar1

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

2.20.3. TimeVacationDays::Calendar1

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

Group:	Framework
SubGroup:	Core::Time::Calendar1
Valid:	1
Required:	1

Config-Setting:	<pre>$Self->{'TimeVacationDays::Calendar1'} = {
	  '1' => {
	    '1' => 'New Year\'s Day'
	  },
	  '12' => {
	    '24' => 'Christmas Eve',
	    '25' => 'First Christmas Day',
	    '26' => 'Second Christmas Day',
	    '31' => 'New Year\'s Eve'
	  },
	  '5' => {
	    '1' => 'International Workers\' Day'
	  }
	};</pre>

2.20.4. TimeVacationDaysOneTime::Calendar1

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

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

2.20.5. TimeWorkingHours::Calendar1

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

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

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

2.21. Core::Time::Calendar2

2.21.1. TimeZone::Calendar2Name

Description:	Defines the name of the indicated calendar.

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

2.21.2. TimeZone::Calendar2

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

2.21.3. TimeVacationDays::Calendar2

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

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

2.21.4. TimeVacationDaysOneTime::Calendar2

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

Config-Setting:	<pre>$Self->{'TimeVacationDaysOneTime::Calendar2'} = {
	 '2004' => {
	  '1' => {</pre>

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

2.21.5. TimeWorkingHours::Calendar2

Description:	Defines the hours and week days of the indicated calendar, to count the working time.
Group:	Framework
SubGroup:	Core::Time::Calendar2
Valid:	1
Required:	1
Config-Setting:	

```
$Self->{'TimeWorkingHours::Calendar2'} =  {
 'Fri' => [
   '8',
   '9',
   '10',
   '11',
   '12',
   '13',
   '14',
   '15',
   '16',
   '17',
   '18',
   '19',
   '20'
 ],
 'Mon' => [
   '8',
   '9',
   '10',
   '11',
   '12',
   '13',
   '14',
   '15',
   '16',
   '17',
   '18',
   '19',
   '20'
 ],
 'Sat' => [],
 'Sun' => [],
 'Thu' => [
   '8',
   '9',
   '10',
   '11',
   '12',
   '13',
   '14',
   '15',
   '16',
   '17',
   '18',
   '19',
   '20'
 ],
 'Tue' => [
   '8',
   '9',
   '10',
   '11',
   '12',
```

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

2.22. Core::Time::Calendar3

2.22.1. TimeZone::Calendar3Name

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

2.22.2. TimeZone::Calendar3

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

2.22.3. TimeVacationDays::Calendar3

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

Valid:	1
Required:	1

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

2.22.4. TimeVacationDaysOneTime::Calendar3

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

Config-Setting:	`$Self->{'TimeVacationDaysOneTime::Calendar3'} = {`
	`'2004' => {`
	`'1' => {`
	`'1' => 'test'`
	`}`
	`}`
	`};`

2.22.5. TimeWorkingHours::Calendar3

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

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

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

2.23. Core::Time::Calendar4

2.23.1. TimeZone::Calendar4Name

Description:	Defines the name of the indicated calendar.
Group:	Framework

SubGroup:	Core::Time::Calendar4
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar4Name'} =  'Calendar Name 4';`

2.23.2. TimeZone::Calendar4

Description:	Defines the time zone of the indicated calendar, which can be assigned later to a specific queue.
Group:	Framework
SubGroup:	Core::Time::Calendar4
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar4'} =  '0';`

2.23.3. TimeVacationDays::Calendar4

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

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

2.23.4. TimeVacationDaysOneTime::Calendar4

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

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

```
    }
  }
};
```

2.23.5. TimeWorkingHours::Calendar4

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

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

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

2.24. Core::Time::Calendar5

2.24.1. TimeZone::Calendar5Name

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

2.24.2. TimeZone::Calendar5

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

2.24.3. TimeVacationDays::Calendar5

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

Required:	1

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

2.24.4. TimeVacationDaysOneTime::Calendar5

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

Config-Setting:	<pre>$Self->{'TimeVacationDaysOneTime::Calendar5'} = {
	 '2004' => {
	  '1' => {
	  '1' => 'test'
	  }
	 }
	};</pre>

2.24.5. TimeWorkingHours::Calendar5

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

Config-Setting:	<pre>$Self->{'TimeWorkingHours::Calendar5'} = {
	 'Fri' => [
	  '8',
	  '9',
	  '10',
	  '11',
	  '12',
	  '13',
	  '14',
	  '15',
	  '16',
	  '17',
	  '18',
	  '19',
	  '20'
	],
	 'Mon' => [
	  '8',
	  '9',
	  '10',</pre>

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

2.25. Core::Time::Calendar6

2.25.1. TimeZone::Calendar6Name

Description:	Defines the name of the indicated calendar.
Group:	Framework
SubGroup:	Core::Time::Calendar6

Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar6Name'} =  'Calendar Name 6';`

2.25.2. TimeZone::Calendar6

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

2.25.3. TimeVacationDays::Calendar6

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

Config-Setting:	`$Self->{'TimeVacationDays::Calendar6'} =  {`
	`  '1' => {`
	`    '1' => 'New Year\'s Day'`
	`  },`
	`  '12' => {`
	`    '24' => 'Christmas Eve',`
	`    '25' => 'First Christmas Day',`
	`    '26' => 'Second Christmas Day',`
	`    '31' => 'New Year\'s Eve'`
	`  },`
	`  '5' => {`
	`    '1' => 'International Workers\' Day'`
	`  }`
	`};`

2.25.4. TimeVacationDaysOneTime::Calendar6

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

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

```
};
```

2.25.5. TimeWorkingHours::Calendar6

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

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

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

2.26. Core::Time::Calendar7

2.26.1. TimeZone::Calendar7Name

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

2.26.2. TimeZone::Calendar7

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

2.26.3. TimeVacationDays::Calendar7

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

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

2.26.4. TimeVacationDaysOneTime::Calendar7

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

2.26.5. TimeWorkingHours::Calendar7

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

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

2.27. Core::Time::Calendar8

2.27.1. TimeZone::Calendar8Name

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

Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar8Name'} =  'Calendar Name 8';`

2.27.2. TimeZone::Calendar8

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

2.27.3. TimeVacationDays::Calendar8

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

Config-Setting:	`$Self->{'TimeVacationDays::Calendar8'} =  {`
	`  '1' => {`
	`    '1' => 'New Year\'s Day'`
	`  },`
	`  '12' => {`
	`    '24' => 'Christmas Eve',`
	`    '25' => 'First Christmas Day',`
	`    '26' => 'Second Christmas Day',`
	`    '31' => 'New Year\'s Eve'`
	`  },`
	`  '5' => {`
	`    '1' => 'International Workers\' Day'`
	`  }`
	`};`

2.27.4. TimeVacationDaysOneTime::Calendar8

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

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

2.27.5. TimeWorkingHours::Calendar8

Description:	Defines the hours and week days of the indicated calendar, to count the working time.
Group:	Framework
SubGroup:	Core::Time::Calendar8
Valid:	1
Required:	1
Config-Setting:	

```
$Self->{'TimeWorkingHours::Calendar8'} =  {
 'Fri' => [
   '8',
   '9',
   '10',
   '11',
   '12',
   '13',
   '14',
   '15',
   '16',
   '17',
   '18',
   '19',
   '20'
 ],
 'Mon' => [
   '8',
   '9',
   '10',
   '11',
   '12',
   '13',
   '14',
   '15',
   '16',
   '17',
   '18',
   '19',
   '20'
 ],
 'Sat' => [],
 'Sun' => [],
 'Thu' => [
   '8',
   '9',
   '10',
   '11',
   '12',
   '13',
   '14',
   '15',
   '16',
   '17',
   '18',
   '19',
   '20'
 ],
 'Tue' => [
   '8',
   '9',
   '10',
   '11',
   '12',
   '13',
   '14',
   '15',
   '16',
   '17',
   '18',
```

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

2.28. Core::Time::Calendar9

2.28.1. TimeZone::Calendar9Name

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

2.28.2. TimeZone::Calendar9

Description:	Defines the time zone of the indicated calendar, which can be assigned later to a specific queue.
Group:	Framework
SubGroup:	Core::Time::Calendar9
Valid:	1
Required:	1
Config-Setting:	<pre>$Self->{'TimeZone::Calendar9'} = '0';</pre>

2.28.3. TimeVacationDays::Calendar9

Description:	Adds the permanent vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar9
Valid:	1
Required:	1

Config-Setting:	<pre>$Self->{'TimeVacationDays::Calendar9'} = {
	'1' => {
	'1' => 'New Year\'s Day'
	},
	'12' => {
	'24' => 'Christmas Eve',</pre>

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

2.28.4. TimeVacationDaysOneTime::Calendar9

Description:	Adds the one time vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar9
Valid:	1
Required:	1

Config-Setting:	`$Self->{'TimeVacationDaysOneTime::Calendar9'} = {`
	`  '2004' => {`
	`    '1' => {`
	`      '1' => 'test'`
	`    }`
	`  }`
	`};`

2.28.5. TimeWorkingHours::Calendar9

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

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

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

2.29. Core::Web

2.29.1. Frontend::WebPath

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

2.29.2. Frontend::ImagePath

Description:	Defines the URL image path of icons for navigation.

Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::ImagePath'} = '<OTRS_CONFIG_Frontend::WebPath>skins/Agent/default/img/';`

2.29.3. Frontend::CSSPath

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

2.29.4. Frontend::JavaScriptPath

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

2.29.5. Frontend::RichText

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

2.29.6. Frontend::RichTextPath

Description:	Defines the URL rich text editor path.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::RichTextPath'} = '<OTRS_CONFIG_Frontend::WebPath>js/thirdparty/ckeditor-4.4.5/';`

2.29.7. Frontend::RichTextWidth

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

2.29.8. Frontend::RichTextHeight

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

2.29.9. Frontend::RichText::DefaultCSS

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

2.29.10. Frontend::RichText::EnhancedMode

Description:	Defines if the enhanced mode should be used (enables use of table, replace, subscript, superscript, paste from word, etc.).
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::RichText::EnhancedMode'} = '0';`

2.29.11. DisableMSIFrameSecurityRestricted

Description:	Disable restricted security for IFrames in IE. May be required for SSO to work in IE8.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	0

Config-Setting:	`$Self->{'DisableMSIFrameSecurityRestricted'} = '0';`

2.29.12. DisableContentSecurityPolicy

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

2.29.13. DefaultViewNewLine

Description:	Automated line break in text messages after x number of chars.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'DefaultViewNewLine'} = '90';`

2.29.14. DefaultViewLines

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

2.29.15. Frontend::AnimationEnabled

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

2.29.16. AttachmentDownloadType

Description:	Allows choosing between showing the attachments of a ticket in the browser (inline) or just make them downloadable (attachment).

Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'AttachmentDownloadType'} = 'attachment';`

2.29.17. WebMaxFileUpload

Description:	Defines the maximal size (in bytes) for file uploads via the browser. Warning: Setting this option to a value which is too low could cause many masks in your OTRS instance to stop working (probably any mask which takes input from the user).
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'WebMaxFileUpload'} = '24000000';`

2.29.18. WebUploadCacheModule

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

2.29.19. Frontend::Output::FilterText###AAAURL

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

2.29.20. Frontend::Themes

Description:	Activates the available themes on the system. Value 1 means active, 0 means inactive.
Group:	Framework
SubGroup:	Core::Web

Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Themes'} = {`
	`'Lite' => '0',`
	`'Standard' => '1'`
	`};`

2.29.21. Frontend::Output::FilterText###OutputFilterTextAutoLink

Description:	Defines a filter to process the text in the articles, in order to highlight predefined keywords.
Group:	Framework
SubGroup:	Core::Web
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Frontend::Output::FilterText'}->{'OutputFilterTextAutoLink'} = {`
	`'Module' => 'Kernel::Output::HTML::OutputFilterTextAutoLink',`
	`'Templates' => {`
	`'AgentTicketZoom' => '1'`
	`}`
	`};`

2.29.22. Frontend::Output::OutputFilterTextAutoLink###CVE

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

Config-Setting:	`$Self->{'Frontend::Output::OutputFilterTextAutoLink'}->{'CVE'} = {`
	`'RegExp' => [`
	`'(CVE|CAN)\\-(\\d{3,4})\\-(\\d{2,})'`
	`],`
	`'Templates' => {`
	`'AgentTicketZoom' => '1'`
	`},`
	`'URL1' => {`
	`'Description' => 'Mitre',`
	`'Image' => 'http://cve.mitre.org/favicon.ico',`
	`'Target' => '_blank',`
	`'URL' => 'http://cve.mitre.org/cgi-bin/cvename.cgi?name=<MATCH1>-<MATCH2>-<MATCH3>'`
	`},`
	`'URL2' => {`
	`'Description' => 'Google',`
	`'Image' => 'http://www.google.de/favicon.ico',`
	`'Target' => '_blank',`
	`'URL' => 'http://google.com/search?q=<MATCH1>-<MATCH2>-<MATCH3>'`
	`},`
	`'URL3' => {`
	`'Description' => 'US-CERT NVD',`
	`'Image' => 'http://nvd.nist.gov/favicon.ico',`

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

2.29.23. Frontend::Output::OutputFilterTextAutoLink###Bugtraq

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

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

2.29.24. Frontend::Output::OutputFilterTextAutoLink###MSBulletins

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

```
 $Self->{'Frontend::Output::OutputFilterTextAutoLink'}-
>{'MSBulletins'} =  {
  'RegExp' => [
    'MS[^A-Za-z]{0,5}(\\d\\d).?(\\d{2,4})'
  ],
```

```
  'Templates' => {
    'AgentTicketZoom' => '1'
  },
  'URL1' => {
    'Description' => 'Microsoft Technet',
    'Image' => 'http://www.microsoft.com/favicon.ico',
    'Target' => '_blank',
    'URL' => 'http://www.microsoft.com/technet/security/bulletin/
MS<MATCH1>-<MATCH2>.mspx'
  },
  'URL2' => {
    'Description' => 'Google',
    'Image' => 'http://www.google.de/favicon.ico',
    'Target' => '_blank',
    'URL' => 'http://google.com/search?q=MS<MATCH1>-<MATCH2>'
  }
};
```

2.29.25. Frontend::Output::OutputFilterTextAutoLink###Setting1

Description:	Define a filter for html output to add links behind a defined string. The element Image allows two input kinds. At once the name of an image (e.g. faq.png). In this case the OTRS image path will be used. The second possiblity is to insert the link to the image.
Group:	Framework
SubGroup:	Core::Web
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Frontend::Output::OutputFilterTextAutoLink'}->{'Setting1'} = {`

```
  'RegExp' => [
    'RegExp'
  ],
  'Templates' => {
    'AgentTicketZoom' => '1'
  },
  'URL1' => {
    'Description' => 'Description',
    'Image' => 'right-small.png',
    'Target' => '_blank',
    'URL' => 'URL'
  },
  'URL2' => {
    'Description' => 'Description',
    'Image' => 'Image',
    'Target' => '_blank',
    'URL' => 'URL'
  }
};
```

2.29.26. Frontend::Output::OutputFilterTextAutoLink###Setting2

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

Required:	0

Config-Setting:	<pre>$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'
	  }
	};</pre>

2.29.27. Loader::Enabled::CSS

Description:	If enabled, OTRS will deliver all CSS files in minified form. WARNING: If you turn this off, there will likely be problems in IE 7, because it cannot load more than 32 CSS files.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<pre>$Self->{'Loader::Enabled::CSS'} = '1';</pre>

2.29.28. Loader::Enabled::JS

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

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

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

Required:	1

Config-Setting:	<pre>$Self->{'Loader::Agent::CommonCSS'}->{'000-Framework'} = [
	  'Core.Reset.css',
	  'Core.Default.css',
	  'Core.Header.css',
	  'Core.OverviewControl.css',
	  'Core.OverviewSmall.css',
	  'Core.OverviewMedium.css',
	  'Core.OverviewLarge.css',
	  'Core.Footer.css',
	  'Core.PageLayout.css',
	  'Core.Form.css',
	  'Core.Table.css',
	  'Core.Widget.css',
	  'Core.WidgetMenu.css',
	  'Core.TicketDetail.css',
	  'Core.Tooltip.css',
	  'Core.Dialog.css',
	  'Core.Print.css',
	  'thirdparty/fontawesome/font-awesome.css'
	];</pre>

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

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

Config-Setting:	<pre>$Self->{'Loader::Agent::CommonCSS::IE8'}->{'000-Framework'} = [
	  'Core.OverviewSmall.IE8.css'
	];</pre>

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

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

Config-Setting:	<pre>$Self->{'Loader::Agent::CommonJS'}->{'000-Framework'} = [
	  'thirdparty/jquery-1.10.0/jquery.js',
	  'thirdparty/jquery-browser-detection/jquery-browser-detection.js',
	  'thirdparty/jquery-ui-1.10.3/jquery-ui.js',
	  'thirdparty/jquery-validate-1.11.1/jquery.validate.js',
	  'thirdparty/stacktrace-0.4/stacktrace.js',
	  'thirdparty/jquery-pubsub/pubsub.js',
	  'thirdparty/jquery-jstree-v.pre1.0/jquery.jstree.js',
	  'thirdparty/jquery-jstree-v.pre1.0/_lib/jquery.hotkeys.js',
	  'Core.JavaScriptEnhancements.js',
	  'Core.Debug.js',
	  'Core.Data.js',
	  'Core.Config.js',
	  'Core.Exception.js',
	  'Core.JSON.js',
	  'Core.App.js',
	  'Core.AJAX.js',
	  'Core.UI.js',</pre>

```
  'Core.UI.Accordion.js',
  'Core.UI.Datepicker.js',
  'Core.UI.Resizable.js',
  'Core.UI.Table.js',
  'Core.UI.Accessibility.js',
  'Core.UI.RichTextEditor.js',
  'Core.UI.Dialog.js',
  'Core.UI.ActionRow.js',
  'Core.UI.Popup.js',
  'Core.UI.TreeSelection.js',
  'Core.UI.Autocomplete.js',
  'Core.Form.js',
  'Core.Form.ErrorTooltips.js',
  'Core.Form.Validate.js',
  'Core.Agent.js',
  'Core.Agent.Search.js',
  'Core.Agent.CustomerInformationCenterSearch.js'
];
```

2.29.32. Loader::Agent::CommonJS###001-JQueryMigrate

Description:	List of JS files to always be loaded for the agent interface.
Group:	Framework
SubGroup:	Core::Web
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Loader::Agent::CommonJS'}->{'001-JQueryMigrate'} = [ 'thirdparty/jquery-migrate-1.2.1/jquery-migrate.js' ];`

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

Description:	List of CSS files to always be loaded for the customer interface.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Loader::Customer::CommonCSS'}->{'000-Framework'} = [ 'Core.Reset.css', 'Core.Default.css', 'Core.Form.css', 'Core.Dialog.css', 'Core.Tooltip.css', 'Core.Login.css', 'Core.Control.css', 'Core.Table.css', 'Core.TicketZoom.css', 'Core.Print.css', 'thirdparty/fontawesome/font-awesome.css' ];`

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

Description:	List of IE8-specific CSS files to always be loaded for the customer interface.

Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Loader::Customer::CommonCSS::IE8'}->{'000-Framework'} = [];`

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

Description:	List of JS files to always be loaded for the customer interface.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Loader::Customer::CommonJS'}->{'000-Framework'} = ['thirdparty/jquery-1.10.0/jquery.js', 'thirdparty/jquery-browser-detection/jquery-browser-detection.js', 'thirdparty/jquery-validate-1.11.1/jquery.validate.js', 'thirdparty/jquery-ui-1.10.3/jquery-ui.js', 'thirdparty/stacktrace-0.4/stacktrace.js', 'thirdparty/jquery-pubsub/pubsub.js', 'thirdparty/jquery-jstree-v.pre1.0/jquery.jstree.js', 'thirdparty/jquery-jstree-v.pre1.0/_lib/jquery.hotkeys.js', 'Core.Debug.js', 'Core.Data.js', 'Core.Exception.js', 'Core.JSON.js', 'Core.JavaScriptEnhancements.js', 'Core.Config.js', 'Core.App.js', 'Core.AJAX.js', 'Core.UI.js', 'Core.UI.Accessibility.js', 'Core.UI.Dialog.js', 'Core.UI.RichTextEditor.js', 'Core.UI.Datepicker.js', 'Core.UI.Popup.js', 'Core.UI.TreeSelection.js', 'Core.UI.Autocomplete.js', 'Core.Form.js', 'Core.Form.ErrorTooltips.js', 'Core.Form.Validate.js', 'Core.Customer.js'];`

2.29.36. Loader::Customer::CommonJS###001-JQueryMigrate

Description:	List of JS files to always be loaded for the agent interface.
Group:	Framework
SubGroup:	Core::Web
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Loader::Customer::CommonJS'}->{'001-JQueryMigrate'} = [ 'thirdparty/jquery-migrate-1.2.1/jquery-migrate.js' ];`

2.29.37. Loader::Agent::DefaultSelectedSkin

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

2.29.38. Loader::Customer::SelectedSkin::HostBased

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

2.30. Core::WebUserAgent

2.30.1. WebUserAgent::Timeout

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

2.30.2. WebUserAgent::Proxy

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

2.30.3. WebUserAgent::DisableSSLVerification

Description:	Turns off SSL certificate validation, for example if you use a transparent HTTPS proxy. Use at your own risk!
Group:	Framework
SubGroup:	Core::WebUserAgent
Valid:	0
Required:	0
Config-Setting:	`$Self->{'WebUserAgent::DisableSSLVerification'} = '0';`

2.31. Crypt::PGP

2.31.1. PGP

Description:	Enables PGP support. When PGP support is enabled for signing and encrypting mail, it is HIGHLY recommended that the web server runs as the OTRS user. Otherwise, there will be problems with the privileges when accessing .gnupg folder.
Group:	Framework
SubGroup:	Crypt::PGP
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PGP'} = '0';`

2.31.2. PGP::Bin

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

2.31.3. PGP::Options

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

2.31.4. PGP::Key::Password

Description:	Sets the password for private PGP key.

Group:	Framework
SubGroup:	Crypt::PGP
Valid:	1
Required:	1

Config-Setting:	`$Self->{'PGP::Key::Password'} = {`
	`'488A0B8F' => 'SomePassword',`
	`'D2DF79FA' => 'SomePassword'`
	`};`

2.31.5. PGP::TrustedNetwork

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

2.31.6. PGP::Log

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

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

2.31.7. PGP::StoreDecryptedData

Description:	If this option is enabled, then the decrypted data will be stored in the database if they are displayed in AgentTicketZoom.
Group:	Framework
SubGroup:	Crypt::PGP
Valid:	1

Required:	1
Config-Setting:	`$Self->{'PGP::StoreDecryptedData'} = '1';`

2.32. Crypt::SMIME

2.32.1. SMIME

Description:	Enables S/MIME support.
Group:	Framework
SubGroup:	Crypt::SMIME
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SMIME'} = '0';`

2.32.2. SMIME::Bin

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

2.32.3. SMIME::CertPath

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

2.32.4. SMIME::PrivatePath

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

2.32.5. SMIME::CacheTTL

Description:	Cache time in seconds for the SSL certificate attributes.

Group:	Framework
SubGroup:	Crypt::SMIME
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SMIME::CacheTTL'} = '86400';`

2.32.6. SMIME::StoreDecryptedData

Description:	If this option is enabled, then the decrypted data will be stored in the database if they are displayed in AgentTicketZoom.
Group:	Framework
SubGroup:	Crypt::SMIME
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SMIME::StoreDecryptedData'} = '1';`

2.33. CustomerInformationCenter

2.33.1. AgentCustomerInformationCenter::MainMenu###010-EditCustomerID

Description:	Main menu registration.
Group:	Framework
SubGroup:	CustomerInformationCenter
Valid:	0
Required:	0
Config-Setting:	`$Self->{'AgentCustomerInformationCenter::MainMenu'}->{'010-EditCustomerID'} = { 'Link' => '$Env{"Baselink"}Action=AdminCustomerCompany;Subaction=Change;CustomerID=$LQData{"CustomerID"};Nav=0', 'Name' => 'Edit customer company' };`

2.34. Frontend::Admin

2.34.1. Events###Package

Description:	List of all Package events to be displayed in the GUI.
Group:	Framework
SubGroup:	Frontend::Admin
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Events'}->{'Package'} = [ 'PackageInstall', 'PackageReinstall', 'PackageUpgrade',`

	'PackageUninstall'];

2.34.2. Events###CustomerUser

Description:	List of all CustomerUser events to be displayed in the GUI.
Group:	Framework
SubGroup:	Frontend::Admin
Valid:	1
Required:	1
Config-Setting:	$Self->{'Events'}->{'CustomerUser'} = ['CustomerUserAdd', 'CustomerUserUpdate'];

2.34.3. Events###CustomerCompany

Description:	List of all CustomerCompany events to be displayed in the GUI.
Group:	Framework
SubGroup:	Frontend::Admin
Valid:	1
Required:	1
Config-Setting:	$Self->{'Events'}->{'CustomerCompany'} = ['CustomerCompanyAdd', 'CustomerCompanyUpdate'];

2.35. Frontend::Admin::AdminCustomerUser

2.35.1. AdminCustomerUser::RunInitialWildcardSearch

Description:	Runs an initial wildcard search of the existing customer users when accessing the AdminCustomerUser module.
Group:	Framework
SubGroup:	Frontend::Admin::AdminCustomerUser
Valid:	1
Required:	1
Config-Setting:	$Self->{'AdminCustomerUser::RunInitialWildcardSearch'} = '1';

2.36. Frontend::Admin::AdminSelectBox

2.36.1. AdminSelectBox::AllowDatabaseModification

Description:	Controls if the admin is allowed to make changes to the database via AdminSelectBox.
Group:	Framework
SubGroup:	Frontend::Admin::AdminSelectBox
Valid:	1

Required:	1
Config-Setting:	`$Self->{'AdminSelectBox::AllowDatabaseModification'} = '0';`

2.37. Frontend::Admin::ModuleRegistration

2.37.1. Frontend::Module###Admin

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

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

2.37.2. Frontend::Module###AdminInit

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

```
$Self->{'Frontend::Module'}->{'AdminInit'} = {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarName' => '',
  'Title' => 'Init'
};
```

2.37.3. Frontend::Module###AdminUser

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

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AdminUser'} = {
	'Description' => 'Create and manage agents.',
	'Group' => [
	'admin'
	],
	'NavBarModule' => {
	'Block' => 'Agent',
	'Description' => 'Create and manage agents.',
	'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
	'Name' => 'Agents',
	'Prio' => '100'
	},
	'NavBarName' => 'Admin',
	'Title' => 'Agents'
	};</pre>

2.37.4. Frontend::Module###AdminGroup

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

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AdminGroup'} = {
	'Description' => 'Admin',
	'Group' => [
	'admin'
	],
	'NavBarModule' => {
	'Block' => 'Agent',
	'Description' => 'Create and manage groups.',
	'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
	'Name' => 'Groups',
	'Prio' => '150'
	},
	'NavBarName' => 'Admin',
	'Title' => 'Groups'
	};</pre>

2.37.5. Frontend::Module###AdminUserGroup

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

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AdminUserGroup'} = {
	'Description' => 'Admin',
	'Group' => [
	'admin'
	],</pre>

```
  'NavBarModule' => {
    'Block' => 'Agent',
    'Description' => 'Link agents to groups.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Agents <-> Groups',
    'Prio' => '200'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Agents <-> Groups'
};
```

2.37.6. Frontend::Module###AdminCustomerUser

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

```
 $Self->{'Frontend::Module'}->{'AdminCustomerUser'} =  {
  'Description' => 'Edit Customer Users',
  'Group' => [
    'admin',
    'users'
  ],
  'GroupRo' => [
    ''
  ],
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.TicketAction.js'
    ]
  },
  'NavBar' => [
    {
      'AccessKey' => '',
      'Block' => 'ItemArea',
      'Description' => 'Customer User Administration.',
      'Link' => 'Action=AdminCustomerUser;Nav=Agent',
      'LinkOption' => '',
      'Name' => 'Customer User Administration',
      'NavBar' => 'Customers',
      'Prio' => '9000',
      'Type' => ''
    }
  ],
  'NavBarModule' => {
    'Block' => 'Customer',
    'Description' => 'Create and manage customer users.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Customer User',
    'Prio' => '300'
  },
  'NavBarName' => 'Customers',
  'Title' => 'Customer Users'
};
```

2.37.7. Frontend::Module###AdminCustomerCompany

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

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AdminCustomerCompany'} = {
	'Description' => 'Admin',
	'Group' => [
	'admin',
	'users'
	],
	'GroupRo' => [
	''
	],
	'NavBar' => [
	{
	'AccessKey' => '',
	'Block' => 'ItemArea',
	'Description' => 'Customer Company Administration.',
	'Link' => 'Action=AdminCustomerCompany;Nav=Agent',
	'LinkOption' => '',
	'Name' => 'Customer Company Administration',
	'NavBar' => 'Customers',
	'Prio' => '9100',
	'Type' => ''
	}
	],
	'NavBarModule' => {
	'Block' => 'Customer',
	'Description' => 'Create and manage customers.',
	'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
	'Name' => 'Customers',
	'Prio' => '310'
	},
	'NavBarName' => 'Admin',
	'Title' => 'Customer Companies'
	};</pre>

2.37.8. Frontend::Module###AdminCustomerUserGroup

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

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

2.37.9. Frontend::Module###AdminCustomerUserService

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

Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AdminCustomerUserService'} = {
	'Description' => 'Admin',
	'Group' => [
	'admin'
	],
	'NavBarModule' => {
	'Block' => 'Customer',
	'Description' => 'Link customer user to services.',
	'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
	'Name' => 'Customer User <-> Services',
	'Prio' => '500'
	},
	'NavBarName' => 'Admin',
	'Title' => 'Customer User <-> Services'
	};</pre>

2.37.10. Frontend::Module###AdminRole

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

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

2.37.11. Frontend::Module###AdminRoleUser

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

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

```
  'Title' => 'Agents <-> Roles'
};
```

2.37.12. Frontend::Module###AdminRoleGroup

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

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

2.37.13. Frontend::Module###AdminSMIME

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

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

2.37.14. Frontend::Module###AdminPGP

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

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

2.37.15. Frontend::Module###AdminMailAccount

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

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

2.37.16. Frontend::Module###AdminPostMasterFilter

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

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

2.37.17. Frontend::Module###AdminEmail

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

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

2.37.18. Frontend::Module###AdminSession

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

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

2.37.19. Frontend::Module###AdminPerformanceLog

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

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AdminPerformanceLog'} = {
	  'Description' => 'Admin',
	  'Group' => [
	    'admin'</pre>

```
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.PerformanceLog.css'
    ]
  },
  'NavBarModule' => {
    'Block' => 'System',
    'Description' => 'View performance benchmark results.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Performance Log',
    'Prio' => '550'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Performance Log'
};
```

2.37.20. Frontend::Module###AdminRegistration

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

```
$Self->{'Frontend::Module'}->{'AdminRegistration'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.Registration.css'
    ]
  },
  'NavBarModule' => {
    'Block' => 'System',
    'Description' => 'Manage system registration.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'System Registration',
    'Prio' => '350'
  },
  'NavBarName' => 'Admin',
  'Title' => 'System Registration'
};
```

2.37.21. Frontend::Module###AdminServiceCenter

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

```
$Self->{'Frontend::Module'}->{'AdminServiceCenter'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.ServiceCenter.css'
    ]
  },
```

```
  'NavBarModule' => {
    'Block' => 'System',
    'Description' => 'Manage OTRS Group services.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Service Center',
    'Prio' => '351'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Service Center'
};
```

2.37.22. Frontend::Module###AdminLog

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

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

2.37.23. Frontend::Module###AdminSelectBox

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

```
 $Self->{'Frontend::Module'}->{'AdminSelectBox'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'System',
    'Description' => 'Execute SQL statements.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'SQL Box',
    'Prio' => '700'
  },
  'NavBarName' => 'Admin',
  'Title' => 'SQL Box'
};
```

2.37.24. Frontend::Module###AdminPackageManager

Description:	Frontend module registration for the agent interface.

Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AdminPackageManager'} = {
	'Description' => 'Software Package Manager',
	'Group' => [
	'admin'
	],
	'NavBarModule' => {
	'Block' => 'System',
	'Description' => 'Update and extend your system with software packages.',
	'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
	'Name' => 'Package Manager',
	'Prio' => '1000'
	},
	'NavBarName' => 'Admin',
	'Title' => 'Package Manager'
	};</pre>

2.38. Frontend::Agent

2.38.1. AgentLogo

Description:	The logo shown in the header of the agent interface. The URL to the image can be a relative URL to the skin image directory, or a full URL to a remote web server.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	0

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

2.38.2. AgentLoginLogo

Description:	The logo shown on top of the login box of the agent interface. The URL to the image must be relative URL to the skin image directory.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	0
Required:	0

Config-Setting:	<pre>$Self->{'AgentLoginLogo'} = {
	'StyleHeight' => '100px',
	'URL' => 'skins/Agent/default/img/loginlogo_default.png'
	};</pre>

2.38.3. LoginURL

Description:	Defines an alternate URL, where the login link refers to.

Group:	Framework
SubGroup:	Frontend::Agent
Valid:	0
Required:	0
Config-Setting:	`$Self->{'LoginURL'} = 'http://host.example.com/login.html';`

2.38.4. LogoutURL

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

2.38.5. PreApplicationModule###AgentInfo

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

2.38.6. InfoKey

Description:	Defines the key to be checked with Kernel::Modules::AgentInfo module. If this user preferences key is true, the message is accepted by the system.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'InfoKey'} = 'wpt22';`

2.38.7. InfoFile

Description:	File that is displayed in the Kernel::Modules::AgentInfo module, if located under Kernel/Output/HTML/Standard/AgentInfo.dtl.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'InfoFile'} = 'AgentInfo';`

2.38.8. LostPassword

Description:	Activates lost password feature for agents, in the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'LostPassword'} =  '1';`

2.38.9. ShowMotd

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

2.38.10. NotificationSubjectLostPasswordToken

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

2.38.11. NotificationBodyLostPasswordToken

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

Config-Setting:	`$Self->{'NotificationBodyLostPasswordToken'} =  'Hi <OTRS_USERFIRSTNAME>,`
	
	`You or someone impersonating you has requested to change your OTRS password.`
	
	`If you want to do this, click on the link below. You will receive another email containing the password.`
	
	`<OTRS_CONFIG_HttpType>://<OTRS_CONFIG_FQDN>/ <OTRS_CONFIG_ScriptAlias>index.pl? Action=LostPassword;Token=<OTRS_TOKEN>`

	If you did not request a new password, please ignore this email. ';

2.38.12. NotificationSubjectLostPassword

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

2.38.13. NotificationBodyLostPassword

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

2.38.14. OpenMainMenuOnHover

Description:	If enabled, the first level of the main menu opens on mouse hover (instead of click only).
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	$Self->{'OpenMainMenuOnHover'} = '0';

2.38.15. FirstnameLastnameOrder

Description:	Specifies the order in which the firstname and the lastname of agents will be displayed.
Group:	Framework
SubGroup:	Frontend::Agent

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

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

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

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

Description:	Default skin for the agent interface (slim version).
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Loader::Agent::Skin'}->{'001-slim'} = { 'Description' => '"Slim" skin which tries to save screen space for power users.', 'HomePage' => 'www.otrs.org', 'InternalName' => 'slim', 'VisibleName' => 'Default (Slim)' };`

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

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

2.38.19. Loader::Agent::Skin###001-ivory-slim

Description:	Balanced white skin by Felix Niklas (slim version).
Group:	Framework
SubGroup:	Frontend::Agent

Valid:	1
Required:	0

Config-Setting:	`$Self->{'Loader::Agent::Skin'}->{'001-ivory-slim'} = {`
	`'Description' => 'Balanced white skin by Felix Niklas (slim version).',`
	`'HomePage' => 'www.felixniklas.de',`
	`'InternalName' => 'ivory-slim',`
	`'VisibleName' => 'Ivory (Slim)'`
	`};`

2.38.20. Loader::Agent::DefaultSelectedSkin::HostBased

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

Config-Setting:	`$Self->{'Loader::Agent::DefaultSelectedSkin::HostBased'} = {`
	`'host1\\.example\\.com' => 'SomeSkin1',`
	`'host2\\.example\\.com' => 'SomeSkin2'`
	`};`

2.38.21. AutoComplete::Agent###Default

Description:	Defines the config options for the autocompletion feature.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	0

Config-Setting:	`$Self->{'AutoComplete::Agent'}->{'Default'} = {`
	`'AutoCompleteActive' => '1',`
	`'ButtonText' => 'Search',`
	`'MaxResultsDisplayed' => '20',`
	`'MinQueryLength' => '2',`
	`'QueryDelay' => '100'`
	`};`

2.38.22. AutoComplete::Agent###CustomerSearch

Description:	Defines the config options for the autocompletion feature.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	0
Config-Setting:	`$Self->{'AutoComplete::Agent'}->{'CustomerSearch'} = {` `'AutoCompleteActive' => '1',`

```
  'ButtonText' => 'Search Customer',
  'MaxResultsDisplayed' => '20',
  'MinQueryLength' => '2',
  'QueryDelay' => '100'
};
```

2.38.23. AutoComplete::Agent###UserSearch

Description:	Defines the config options for the autocompletion feature.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	0

Config-Setting:	`$Self->{'AutoComplete::Agent'}->{'UserSearch'} = {`
	`'AutoCompleteActive' => '1',`
	`'ButtonText' => 'Search User',`
	`'MaxResultsDisplayed' => '20',`
	`'MinQueryLength' => '2',`
	`'QueryDelay' => '100'`
	`};`

2.38.24. PossibleNextActions

Description:	Defines the list of possible next actions on an error screen.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	0

Config-Setting:	`$Self->{'PossibleNextActions'} = {`
	`'Go to dashboard!' => '$Env{"CGIHandle"}?Action=AgentDashboard'`
	`};`

2.39. Frontend::Agent::Dashboard

2.39.1. AgentCustomerInformationCenter::Backend###0600-CIC-CustomerCompanyInformation

Description:	Parameters for the dashboard backend of the customer company information of the agent interface . "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0

Config-Setting:	`$Self->{'AgentCustomerInformationCenter::Backend'}->{'0600-CIC-CustomerCompanyInformation'} = {`
	`'Attributes' => '',`
	`'Block' => 'ContentSmall',`
	`'Default' => '1',`
	`'Description' => 'Customer Company Information',`

```
  'Group' => '',
  'Module' =>
 'Kernel::Output::HTML::DashboardCustomerCompanyInformation',
  'Title' => 'Customer Company Information'
};
```

2.39.2. DashboardBackend###0000-ProductNotify

Description:	Defines the parameters for the dashboard backend. "Group" is used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" indicates if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" defines the cache expiration period in minutes for the plugin.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0

Config-Setting:	`$Self->{'DashboardBackend'}->{'0000-ProductNotify'} = {`
	`'Block' => 'ContentLarge',`
	`'CacheTTLLocal' => '1440',`
	`'Default' => '1',`
	`'Description' => 'News about OTRS releases!',`
	`'Group' => 'admin',`
	`'Module' => 'Kernel::Output::HTML::DashboardProductNotify',`
	`'Title' => 'Product News',`
	`'URL' => 'http://www.otrs.com/product.xml'`
	`};`

2.39.3. DashboardBackend###0390-UserOutOfOffice

Description:	Defines the parameters for the dashboard backend. "Limit" defines the number of entries displayed by default. "Group" is used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" indicates if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" defines the cache expiration period in minutes for the plugin.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0

Config-Setting:	`$Self->{'DashboardBackend'}->{'0390-UserOutOfOffice'} = {`
	`'Block' => 'ContentSmall',`
	`'CacheTTLLocal' => '5',`
	`'Default' => '1',`
	`'Description' => '',`
	`'Group' => '',`
	`'IdleMinutes' => '60',`
	`'Limit' => '10',`
	`'Module' => 'Kernel::Output::HTML::DashboardUserOutOfOffice',`
	`'SortBy' => 'UserFullname',`
	`'Title' => 'Out Of Office'`
	`};`

2.39.4. DashboardBackend###0400-UserOnline

Description:	Defines the parameters for the dashboard backend. "Limit" defines the number of entries displayed by default. "Group" is used to

	restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" indicates if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" defines the cache expiration period in minutes for the plugin.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0

Config-Setting:	`$Self->{'DashboardBackend'}->{'0400-UserOnline'} = {`
	`'Block' => 'ContentSmall',`
	`'CacheTTLLocal' => '5',`
	`'Default' => '0',`
	`'Description' => '',`
	`'Filter' => 'Agent',`
	`'Group' => '',`
	`'IdleMinutes' => '60',`
	`'Limit' => '10',`
	`'Module' => 'Kernel::Output::HTML::DashboardUserOnline',`
	`'ShowEmail' => '0',`
	`'SortBy' => 'UserFullname',`
	`'Title' => 'Online'`
	`};`

2.39.5. DashboardBackend###0410-RSS

Description:	Defines the parameters for the dashboard backend. "Limit" defines the number of entries displayed by default. "Group" is used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" indicates if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTL" indicates the cache expiration period in minutes for the plugin.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0

Config-Setting:	`$Self->{'DashboardBackend'}->{'0410-RSS'} = {`
	`'Block' => 'ContentSmall',`
	`'CacheTTL' => '360',`
	`'Default' => '1',`
	`'Description' => '',`
	`'Group' => '',`
	`'Limit' => '6',`
	`'Module' => 'Kernel::Output::HTML::DashboardRSS',`
	`'Title' => 'OTRS News',`
	`'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'`
	`};`

2.39.6. DashboardBackend###0200-Image

Description:	Defines the parameters for the dashboard backend. "Group" is used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" indicates if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTL" indicates the cache expiration period in minutes for the plugin.

Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	0
Required:	0

Config-Setting:	<pre>$Self->{'DashboardBackend'}->{'0200-Image'} = {
	'Block' => 'ContentLarge',
	'Default' => '1',
	'Description' => 'Some picture description!',
	'Group' => '',
	'Height' => '140',
	'Link' => 'http://otrs.org/',
	'LinkTitle' => 'http://otrs.org/',
	'Module' => 'Kernel::Output::HTML::DashboardImage',
	'Title' => 'A picture',
	'URL' => 'http://www.otrs.com/uploads/pics/
	jointhecommunity_02.jpg',
	'Width' => '198'
	};</pre>

2.39.7. DashboardBackend###0210-MOTD

Description:	Shows the message of the day (MOTD) in the agent dashboard. "Group" is used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" indicates if the plugin is enabled by default or if the user needs to enable it manually.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	0
Required:	0

Config-Setting:	<pre>$Self->{'DashboardBackend'}->{'0210-MOTD'} = {
	'Block' => 'ContentLarge',
	'Default' => '1',
	'Group' => '',
	'Module' => 'Kernel::Output::HTML::DashboardMOTD',
	'Title' => 'Message of the Day'
	};</pre>

2.39.8. DashboardBackend###0300-IFrame

Description:	Defines the parameters for the dashboard backend. "Group" is used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" indicates if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTL" indicates the cache expiration period in minutes for the plugin.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	0
Required:	0

Config-Setting:	<pre>$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/',</pre>

	'Marginheight' => '5', 'Marginwidth' => '5', 'Module' => 'Kernel::Output::HTML::DashboardIFrame', 'Scrolling' => 'auto', 'Title' => 'A Website', 'URL' => 'http://www.otrs.org/', 'Width' => '1024' };

2.39.9. AgentCustomerInformationCenter::Backend###0050-CIC-CustomerUserList

Description:	Parameters for the dashboard backend of the customer user list overview of the agent interface . "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0

Config-Setting:	$Self->{'AgentCustomerInformationCenter::Backend'}->{'0050-CIC-CustomerUserList'} = {
	'Attributes' => '',
	'Block' => 'ContentLarge',
	'CacheTTLLocal' => '0.5',
	'Default' => '1',
	'Description' => 'All customer users of a CustomerID',
	'Group' => '',
	'Limit' => '10',
	'Module' => 'Kernel::Output::HTML::DashboardCustomerUserList',
	'Permission' => 'ro',
	'Title' => 'Customer Users'
	};

2.40. Frontend::Agent::LinkObject

2.40.1. Frontend::AgentLinkObject::WildcardSearch

Description:	Starts a wildcard search of the active object after the link object mask is started.
Group:	Framework
SubGroup:	Frontend::Agent::LinkObject
Valid:	1
Required:	1
Config-Setting:	$Self->{'Frontend::AgentLinkObject::WildcardSearch'} = '0';

2.41. Frontend::Agent::ModuleMetaHead

2.41.1. Frontend::HeaderMetaModule###100-Refresh

Description:	Defines the module to generate html refresh headers of html sites.

Group:	Framework
SubGroup:	Frontend::Agent::ModuleMetaHead
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Frontend::HeaderMetaModule'}->{'100-Refresh'} = {`
	`  'Module' => 'Kernel::Output::HTML::HeaderMetaRefresh'`
	`};`

2.42. Frontend::Agent::ModuleNotify

2.42.1. Frontend::NotifyModule###200-UID-Check

Description:	Defines the module to display a notification in the agent interface, if the system is used by the admin user (normally you shouldn't work as admin).
Group:	Framework
SubGroup:	Frontend::Agent::ModuleNotify
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Frontend::NotifyModule'}->{'200-UID-Check'} = {`
	`  'Module' => 'Kernel::Output::HTML::NotificationUIDCheck'`
	`};`

2.42.2. Frontend::NotifyModule###300-ShowAgentOnline

Description:	Defines the module that shows all the currently logged in agents in the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleNotify
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Frontend::NotifyModule'}->{'300-ShowAgentOnline'} = {`
	`  'IdleMinutes' => '60',`
	`  'Module' => 'Kernel::Output::HTML::NotificationAgentOnline',`
	`  'ShowEmail' => '1'`
	`};`

2.42.3. Frontend::NotifyModule###400-ShowCustomerOnline

Description:	Defines the module that shows all the currently loged in customers in the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleNotify
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Frontend::NotifyModule'}->{'400-ShowCustomerOnline'} = {`
	`  'IdleMinutes' => '60',`
	`  'Module' => 'Kernel::Output::HTML::NotificationCustomerOnline',`

	'ShowEmail' => '1' };

2.42.4. Frontend::NotifyModule###500-OutofOffice-Check

Description:	Defines the module to display a notification in the agent interface, if the agent is logged in while having out-of-office active.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleNotify
Valid:	1
Required:	1

Config-Setting:	$Self->{'Frontend::NotifyModule'}->{'500-OutofOffice-Check'} = {
	'Module' => 'Kernel::Output::HTML::NotificationOutofOfficeCheck'
	};

2.42.5. Frontend::NotifyModule###900-Generic

Description:	Defines the module that shows a generic notification in the agent interface. Either "Text" - if configured - or the contents of "File" will be displayed.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleNotify
Valid:	0
Required:	0

Config-Setting:	$Self->{'Frontend::NotifyModule'}->{'900-Generic'} = {
	'File' => '<OTRS_CONFIG_Home>/var/notify.txt',
	'Link' => 'http://www.otrs.com',
	'Module' => 'Kernel::Output::HTML::NotificationGeneric',
	'Priority' => 'Warning',
	'Text' => 'The OTRS Website'
	};

2.43. Frontend::Agent::ModuleRegistration

2.43.1. Frontend::Module###Logout

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

Config-Setting:	$Self->{'Frontend::Module'}->{'Logout'} = {
	'Description' => 'Logout',
	'NavBarName' => '',
	'Title' => ''
	};

2.43.2. Frontend::Module###AgentDashboard

Description:	Frontend module registration for the agent interface.
Group:	Framework

SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AgentDashboard'} = {
	 'Description' => 'Agent Dashboard',
	 'Loader' => {
	  'CSS' => [
	  'Core.Agent.Dashboard.css',
	  'Core.AllocationList.css',
	  'thirdparty/fullcalendar-1.6.1/fullcalendar.css',
	  'thirdparty/d3js/nv.d3.css'
	  ],
	  'JavaScript' => [
	  'thirdparty/flot-0.8.1/excanvas.js',
	  'thirdparty/flot-0.8.1/jquery.flot.js',
	  'thirdparty/fullcalendar-1.6.1/fullcalendar.min.js',
	  'thirdparty/d3js/d3.v3.min.js',
	  'thirdparty/d3js/nv.d3.min.js',
	  'thirdparty/d3js/models/OTRSmultiBarChart.js',
	  'thirdparty/d3js/models/OTRSstackedAreaChart.js',
	  'Core.UI.Chart.js',
	  'Core.UI.AdvancedChart.js',
	  'Core.UI.DnD.js',
	  'Core.UI.AllocationList.js',
	  'Core.Agent.TableFilters.js',
	  'Core.Agent.Dashboard.js'
	  ]
	 },
	 'NavBar' => [
	  {
	  'AccessKey' => 'd',
	  'Block' => 'ItemArea',
	  'Description' => '',
	  'Link' => 'Action=AgentDashboard',
	  'LinkOption' => '',
	  'Name' => 'Dashboard',
	  'NavBar' => 'Dashboard',
	  'Prio' => '50',
	  'Type' => 'Menu'
	  }
	],
	 'NavBarName' => 'Dashboard',
	 'Title' => ''
	};</pre>

2.43.3. Frontend::Module###AgentCustomerInformationCenter

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

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AgentCustomerInformationCenter'} =
	{
	 'Description' => 'Customer Information Center',
	 'Loader' => {
	  'CSS' => [
	  'Core.AllocationList.css'
	  ],
	  'JavaScript' => [
	  'thirdparty/flot-0.8.1/excanvas.js',
	  'thirdparty/flot-0.8.1/jquery.flot.js',
	  'Core.UI.Chart.js',</pre>

```
      'Core.UI.DnD.js',
      'Core.UI.AllocationList.js',
      'Core.Agent.Dashboard.js',
      'Core.Agent.TableFilters.js'
    ]
  },
  'NavBar' => [
    {
      'AccessKey' => '',
      'Block' => 'ItemArea',
      'Description' => 'Customer Information Center.',
      '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' => 'c',
      'Block' => 'ItemArea',
      'Description' => 'Customer Information Center.',
      'Link' => 'Action=AgentCustomerInformationCenter',
      'LinkOption' => '',
      'Name' => 'Customers',
      'NavBar' => 'Customers',
      'Prio' => '60',
      'Type' => 'Menu'
    }
  ],
  'NavBarName' => 'Customer Information Center',
  'Title' => ''
};
```

2.43.4. Frontend::Module###AgentCustomerInformationCenterSearch

Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentCustomerInformationCenterSearch'} = { 'Description' => 'Customer Information Center Search', 'Title' => '' };`

2.43.5. Frontend::Module###AgentPreferences

Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentPreferences'} = { 'Description' => 'Agent Preferences', 'NavBarName' => 'Preferences', 'Title' => ''`

	};

2.43.6. Frontend::Module###PictureUpload

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

Config-Setting:	$Self->{'Frontend::Module'}->{'PictureUpload'} = {
	'Description' => 'Picture upload module',
	'NavBarName' => 'Ticket',
	'Title' => 'Picture-Upload'
	};

2.43.7. Frontend::Module###AgentSpelling

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

Config-Setting:	$Self->{'Frontend::Module'}->{'AgentSpelling'} = {
	'Description' => 'Spell checker',
	'Loader' => {
	'JavaScript' => [
	'Core.Agent.TicketAction.js'
	]
	},
	'NavBarName' => '',
	'Title' => 'Spell Checker'
	};

2.43.8. Frontend::Module###SpellingInline

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

Config-Setting:	$Self->{'Frontend::Module'}->{'SpellingInline'} = {
	'Description' => 'Spell checker',
	'NavBarName' => '',
	'Title' => 'Spell Checker'
	};

2.43.9. Frontend::Module###AgentBook

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

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AgentBook'} = {
	'Description' => 'Address book of CustomerUser sources',
	'Loader' => {
	'JavaScript' => [
	'Core.Agent.CustomerSearch.js',
	'Core.Agent.TicketAction.js'
	]
	},
	'NavBarName' => '',
	'Title' => 'Address Book'
	};</pre>

2.43.10. Frontend::Module###AgentLinkObject

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

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AgentLinkObject'} = {
	'Description' => 'Link Object',
	'NavBarName' => '',
	'Title' => 'Link Object'
	};</pre>

2.43.11. Frontend::Module###AgentInfo

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

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AgentInfo'} = {
	'Description' => 'Generic Info module',
	'NavBarName' => '',
	'Title' => 'Info'
	};</pre>

2.43.12. Frontend::Module###AgentSearch

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

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AgentSearch'} = {
	'Description' => 'Global Search Module',
	'NavBarName' => '',
	'Title' => 'Search'
	};</pre>

2.43.13. CustomerFrontend::Module###SpellingInline

Description:	Frontend module registration for the customer interface.
Group:	Framework

SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'CustomerFrontend::Module'}->{'SpellingInline'} = {`
	`'Description' => 'Spell checker',`
	`'NavBarName' => '',`
	`'Title' => 'Spell Checker'`
	`};`

2.43.14. Frontend::Module###AgentHTMLReference

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

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentHTMLReference'} = {`
	`'Description' => 'HTML Reference',`
	`'Group' => [`
	`'users'`
	`],`
	`'GroupRo' => [`
	`'users'`
	`],`
	`'Loader' => {`
	`'CSS' => [`
	`'Core.Agent.HTMLReference.css'`
	`]`
	`},`
	`'NavBarName' => '',`
	`'Title' => 'HTML Reference'`
	`};`

2.43.15. Frontend::Module###AgentStats

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

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentStats'} = {`
	`'Description' => 'Stats',`
	`'Group' => [`
	`'stats'`
	`],`
	`'GroupRo' => [`
	`'stats'`
	`],`
	`'Loader' => {`
	`'JavaScript' => [`
	`'Core.Agent.Stats.js'`
	`]`
	`},`
	`'NavBar' => [`
	`{`
	`'AccessKey' => '',`
	`'Block' => 'ItemArea',`
	`'Description' => '',`
	`'Link' => 'Action=AgentStats;Subaction=Overview',`
	`'LinkOption' => '',`

```
      'Name' => 'Statistics',
      'NavBar' => 'Stats',
      'Prio' => '8500',
      'Type' => 'Menu'
    },
    {
      'AccessKey' => '',
      'Block' => '',
      'Description' => 'Overview',
      'GroupRo' => [
        'stats'
      ],
      'Link' => 'Action=AgentStats;Subaction=Overview',
      'LinkOption' => '',
      'Name' => 'Overview',
      'NavBar' => 'Stats',
      'Prio' => '100',
      'Type' => ''
    },
    {
      'AccessKey' => '',
      'Block' => '',
      'Description' => 'New',
      'Group' => [
        'stats'
      ],
      'Link' => 'Action=AgentStats;Subaction=Add',
      'LinkOption' => '',
      'Name' => 'New',
      'NavBar' => 'Stats',
      'Prio' => '200',
      'Type' => ''
    },
    {
      'AccessKey' => '',
      'Block' => '',
      'Description' => 'Import',
      'Group' => [
        'stats'
      ],
      'Link' => 'Action=AgentStats;Subaction=Import',
      'LinkOption' => '',
      'Name' => 'Import',
      'NavBar' => 'Stats',
      'Prio' => '300',
      'Type' => ''
    }
  ],
  'NavBarName' => 'Stats',
  'Title' => 'Stats'
};
```

2.44. Frontend::Agent::NavBarModule

2.44.1. Frontend::NavBarModule###6-CustomerCompany

Description:	Frontend module registration (disable company link if no company feature is used).
Group:	Framework
SubGroup:	Frontend::Agent::NavBarModule
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::NavBarModule'}->{'6-CustomerCompany'} = {` `'Module' => 'Kernel::Output::HTML::NavBarCustomerCompany'`

	};

2.45. Frontend::Agent::Preferences

2.45.1. PreferencesTableValue

Description:	Defines the name of the column to store the data in the preferences table.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PreferencesTableValue'} = 'preferences_value';`

2.45.2. PreferencesTableUserID

Description:	Defines the name of the column to store the user identifier in the preferences table.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PreferencesTableUserID'} = 'user_id';`

2.45.3. PreferencesView

Description:	Sets the display order of the different items in the preferences view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PreferencesView'} = [ 'User Profile', 'Email Settings', 'Other Settings' ];`

2.45.4. PreferencesGroups###Password

Description:	Defines the config parameters of this item, to be shown in the preferences view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	`$Self->{'PreferencesGroups'}->{'Password'} = { 'Active' => '1', 'Area' => 'Agent', 'Column' => 'User Profile',`

```
  'Label' => 'Change password',
  'Module' => 'Kernel::Output::HTML::PreferencesPassword',
  'PasswordMaxLoginFailed' => '0',
  'PasswordMin2Characters' => '0',
  'PasswordMin2Lower2UpperCharacters' => '0',
  'PasswordMinSize' => '0',
  'PasswordNeedDigit' => '0',
  'PasswordRegExp' => '',
  'Prio' => '0500'
};
```

2.45.5. PreferencesGroups###SpellDict

Description:	Defines the config parameters of this item, to be shown in the preferences view. Take care to maintain the dictionaries installed in the system in the data section.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	0
Required:	0

Config-Setting:	<pre>$Self->{'PreferencesGroups'}->{'SpellDict'} = {
	  'Active' => '1',
	  'Column' => 'User Profile',
	  'Data' => {
	    'deutsch' => 'Deutsch',
	    'english' => 'English'
	  },
	  'DataSelected' => 'english',
	  'Key' => 'Default spelling dictionary',
	  'Label' => 'Spelling Dictionary',
	  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',
	  'PrefKey' => 'UserSpellDict',
	  'Prio' => '2000'
	};</pre>

2.45.6. PreferencesGroups###Comment

Description:	Defines the config parameters of this item, to be shown in the preferences view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'PreferencesGroups'}->{'Comment'} = {
	  'Active' => '0',
	  'Block' => 'Input',
	  'Column' => 'Other Settings',
	  'Data' => '$Env{"UserComment"}',
	  'Key' => 'Comment',
	  'Label' => 'Comment',
	  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',
	  'PrefKey' => 'UserComment',
	  'Prio' => '6000'
	};</pre>

2.45.7. PreferencesGroups###Language

Description:	Defines the config parameters of this item, to be shown in the preferences view.

Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	`$Self->{'PreferencesGroups'}->{'Language'} = {`
	`'Active' => '1',`
	`'Column' => 'User Profile',`
	`'Key' => 'Frontend language',`
	`'Label' => 'Language',`
	`'Module' => 'Kernel::Output::HTML::PreferencesLanguage',`
	`'PrefKey' => 'UserLanguage',`
	`'Prio' => '1000'`
	`};`

2.45.8. PreferencesGroups###Skin

Description:	Defines the config parameters of this item, to be shown in the preferences view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	`$Self->{'PreferencesGroups'}->{'Skin'} = {`
	`'Active' => '1',`
	`'Column' => 'User Profile',`
	`'Key' => 'Wear this frontend skin',`
	`'Label' => 'Skin',`
	`'Module' => 'Kernel::Output::HTML::PreferencesSkin',`
	`'PrefKey' => 'UserSkin',`
	`'Prio' => '2000'`
	`};`

2.45.9. PreferencesGroups###Theme

Description:	Defines the config parameters of this item, to be shown in the preferences view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	`$Self->{'PreferencesGroups'}->{'Theme'} = {`
	`'Active' => '1',`
	`'Column' => 'User Profile',`
	`'Key' => 'Frontend theme',`
	`'Label' => 'Theme',`
	`'Module' => 'Kernel::Output::HTML::PreferencesTheme',`
	`'PrefKey' => 'UserTheme',`
	`'Prio' => '3000'`
	`};`

2.45.10. PreferencesGroups###OutOfOffice

Description:	Defines the config parameters of this item, to be shown in the preferences view.
Group:	Framework

SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'PreferencesGroups'}->{'OutOfOffice'} = {
	'Active' => '1',
	'Block' => 'OutOfOffice',
	'Column' => 'User Profile',
	'Key' => '',
	'Label' => 'Out Of Office Time',
	'Module' => 'Kernel::Output::HTML::PreferencesOutOfOffice',
	'PrefKey' => 'UserOutOfOffice',
	'Prio' => '4000'
	};</pre>

2.45.11. PreferencesGroups###TimeZone

Description:	Defines the config parameters of this item, to be shown in the preferences view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'PreferencesGroups'}->{'TimeZone'} = {
	'Active' => '1',
	'Column' => 'User Profile',
	'Key' => 'Time Zone',
	'Label' => 'Time Zone',
	'Module' => 'Kernel::Output::HTML::PreferencesTimeZone',
	'PrefKey' => 'UserTimeZone',
	'Prio' => '5000'
	};</pre>

2.45.12. PreferencesGroups###CSVSeparator

Description:	Gives end users the possibility to override the separator character for CSV files, defined in the translation files.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	0
Required:	0

Config-Setting:	<pre>$Self->{'PreferencesGroups'}->{'CSVSeparator'} = {
	'Active' => '1',
	'Column' => 'Other Settings',
	'Data' => {
	'' => '',
	',' => ',',
	';' => ';',
	'\\t' => 'tab',
	'|' => '|'
	},
	'DataSelected' => '0',
	'Desc' => 'Select the separator character used in CSV files (stats and searches). If you don\'t select a separator here, the default separator for your language will be used.',
	'Key' => 'CSV Separator',
	'Label' => 'CSV Separator',
	'Module' => 'Kernel::Output::HTML::PreferencesGeneric',
	'PrefKey' => 'UserCSVSeparator',
	'Prio' => '4000'</pre>

	};

2.46. Frontend::Agent::SearchRouter

2.46.1. Frontend::SearchDefault

Description:	Search backend default router.
Group:	Framework
SubGroup:	Frontend::Agent::SearchRouter
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::SearchDefault'} = 'Action=AgentTicketSearch;Subaction=AJAX';`

2.47. Frontend::Agent::Stats

2.47.1. Stats::SearchPageShown

Description:	Defines the default maximum number of search results shown on the overview page.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::SearchPageShown'} = '20';`

2.47.2. Stats::DefaultSelectedDynamicObject

Description:	Defines the default selection at the drop down menu for dynamic objects (Form: Common Specification).
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::DefaultSelectedDynamicObject'} = 'Ticket';`

2.47.3. Stats::DefaultSelectedPermissions

Description:	Defines the default selection at the drop down menu for permissions (Form: Common Specification).
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::DefaultSelectedPermissions'} = [ 'stats' ];`

2.47.4. Stats::DefaultSelectedFormat

Description:	Defines the default selection at the drop down menu for stats format (Form: Common Specification). Please insert the format key (see Stats::Format).
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::DefaultSelectedFormat'} = [ 'Print', 'CSV' ];`

2.47.5. Stats::SearchLimit

Description:	Defines the search limit for the stats.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::SearchLimit'} = '500';`

2.47.6. Stats::Format

Description:	Defines all the possible stats output formats.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Format'} = { 'CSV' => 'CSV', 'GD::Graph::area' => 'graph-area', 'GD::Graph::bars' => 'graph-bars', 'GD::Graph::hbars' => 'graph-hbars', 'GD::Graph::lines' => 'graph-lines', 'GD::Graph::linespoints' => 'graph-lines-points', 'GD::Graph::pie' => 'graph-pie', 'GD::Graph::points' => 'graph-points', 'Print' => 'Print' };`

2.47.7. Stats::GraphSize

Description:	Sets the size of the statistic graph.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::GraphSize'} = { '1200x800' => '1200x800',`

	`'1600x1200' => '1600x1200', '800x600' => '800x600' };`

2.47.8. Stats::TimeType

Description:	Sets the time type which should be shown.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::TimeType'} = 'Extended';`

2.47.9. Stats::ExchangeAxis

Description:	Allows agents to exchange the axis of a stat if they generate one.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Stats::ExchangeAxis'} = '0';`

2.47.10. Stats::UseAgentElementInStats

Description:	Allows agents to generate individual-related stats.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Stats::UseAgentElementInStats'} = '0';`

2.47.11. Stats::UseInvalidAgentInStats

Description:	Allows invalid agents to generate individual-related stats.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Stats::UseInvalidAgentInStats'} = '1';`

2.47.12. Stats::CustomerIDAsMultiSelect

Description:	Shows all the customer identifiers in a multi-select field (not useful if you have a lot of customer identifiers).
Group:	Framework
SubGroup:	Frontend::Agent::Stats

Valid:	1
Required:	0
Config-Setting:	`$Self->{'Stats::CustomerIDAsMultiSelect'} =  '1';`

2.48. Frontend::Customer

2.48.1. CustomerHeadline

Description:	The headline shown in the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerHeadline'} =  'Example Company Support';`

2.48.2. CustomerLogo

Description:	The logo shown in the header of the customer interface. The URL to the image can be a relative URL to the skin image directory, or a full URL to a remote web server.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0

Config-Setting:	`$Self->{'CustomerLogo'} =  {`
	`  'StyleHeight' => '50px',`
	`  'StyleRight' => '25px',`
	`  'StyleTop' => '2px',`
	`  'StyleWidth' => '135px',`
	`  'URL' => 'skins/Customer/default/img/logo.png'`
	`};`

2.48.3. CustomerPanelUserID

Description:	Defines the user identifier for the customer panel.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerPanelUserID'} =  '1';`

2.48.4. CustomerGroupSupport

Description:	Activates support for customer groups.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1

Config-Setting:	`$Self->{'CustomerGroupSupport'} = '0';`

2.48.5. CustomerGroupAlwaysGroups

Description:	Defines the groups every customer user will be in (if CustomerGroupSupport is enabled and you don't want to manage every user for these groups).
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerGroupAlwaysGroups'} = [ 'users' ];`

2.48.6. CustomerPanelLoginURL

Description:	Defines an alternate login URL for the customer panel..
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	`$Self->{'CustomerPanelLoginURL'} = 'http://host.example.com/cgi-bin/login.pl';`

2.48.7. CustomerPanelLogoutURL

Description:	Defines an alternate logout URL for the customer panel.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	`$Self->{'CustomerPanelLogoutURL'} = 'http://host.example.com/cgi-bin/login.pl';`

2.48.8. Frontend::CustomerUser::Item###1-GoogleMaps

Description:	Defines a customer item, which generates a google maps icon at the end of a customer info block.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::CustomerUser::Item'}->{'1-GoogleMaps'} = { 'Attributes' => 'UserStreet;UserCity;UserCountry;', 'CSS' => 'Core.Agent.CustomerUser.GoogleMaps.css', 'CSSClass' => 'GoogleMaps', 'Module' => 'Kernel::Output::HTML::CustomerUserGeneric',`

```
  'Required' => 'UserStreet;UserCity;',
  'Target' => '_blank',
  'Text' => 'Location',
  'URL' => 'http://maps.google.com/maps?z=7&q='
};
```

2.48.9. Frontend::CustomerUser::Item###2-Google

Description:	Defines a customer item, which generates a google icon at the end of a customer info block.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Frontend::CustomerUser::Item'}->{'2-Google'} = { 'Attributes' => 'UserFirstname;UserLastname;', 'CSS' => 'Core.Agent.CustomerUser.Google.css', 'CSSClass' => 'Google', 'Module' => 'Kernel::Output::HTML::CustomerUserGeneric', 'Required' => 'UserFirstname;UserLastname;', 'Target' => '_blank', 'Text' => 'Google', 'URL' => 'http://google.com/search?q=' };`

2.48.10. Frontend::CustomerUser::Item###2-LinkedIn

Description:	Defines a customer item, which generates a LinkedIn icon at the end of a customer info block.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Frontend::CustomerUser::Item'}->{'2-LinkedIn'} = { 'Attributes' => 'UserFirstname;UserLastname;', 'CSS' => 'Core.Agent.CustomerUser.LinkedIn.css', 'CSSClass' => 'LinkedIn', 'Module' => 'Kernel::Output::HTML::CustomerUserGeneric', 'Required' => 'UserFirstname;UserLastname;', 'Target' => '_blank', 'Text' => 'LinkedIn', 'URL' => 'http://www.linkedin.com/commonSearch?type=people&keywords=' };`

2.48.11. Frontend::CustomerUser::Item###3-XING

Description:	Defines a customer item, which generates a XING icon at the end of a customer info block.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Frontend::CustomerUser::Item'}->{'3-XING'} = { 'Attributes' => 'UserFirstname;UserLastname;', 'CSS' => 'Core.Agent.CustomerUser.Xing.css',`

	'CSSClass' => 'Xing', 'Module' => 'Kernel::Output::HTML::CustomerUserGeneric', 'Required' => 'UserFirstname;UserLastname;', 'Target' => '_blank', 'Text' => 'XING', 'URL' => 'https://www.xing.com/app/search?op=search;keywords=' };

2.48.12. CustomerPanelPreApplicationModule###CustomerAccept

Description:	This module and its PreRun() function will be executed, if defined, for every request. This module is useful to check some user options or to display news about new applications.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	$Self->{'CustomerPanelPreApplicationModule'}->{'CustomerAccept'} = 'Kernel::Modules::CustomerAccept';

2.48.13. CustomerPanel::InfoKey

Description:	Defines the key to check with CustomerAccept. If this user preferences key is true, then the message is accepted by the system.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	$Self->{'CustomerPanel::InfoKey'} = 'CustomerAccept1';

2.48.14. CustomerPanel::InfoFile

Description:	Defines the path of the shown info file, that is located under Kernel/Output/HTML/Standard/CustomerAccept.dtl.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	$Self->{'CustomerPanel::InfoFile'} = 'CustomerAccept';

2.48.15. CustomerPanelLostPassword

Description:	Activates lost password feature for customers.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1

Config-Setting:	`$Self->{'CustomerPanelLostPassword'} = '1';`

2.48.16. CustomerPanelCreateAccount

Description:	Enables customers to create their own accounts.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerPanelCreateAccount'} = '1';`

2.48.17. CustomerPanelSubjectLostPasswordToken

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

2.48.18. CustomerPanelBodyLostPasswordToken

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

Config-Setting:	`$Self->{'CustomerPanelBodyLostPasswordToken'} = 'Hi <OTRS_USERFIRSTNAME>,`
	
	`You or someone impersonating you has requested to change your OTRS password.`
	
	`If you want to do this, click on this link. You will receive another email containing the password.`
	
	`<OTRS_CONFIG_HttpType>://<OTRS_CONFIG_FQDN>/ <OTRS_CONFIG_ScriptAlias>customer.pl? Action=CustomerLostPassword;Token=<OTRS_TOKEN>`
	
	`If you did not request a new password, please ignore this email. ';`

2.48.19. CustomerPanelSubjectLostPassword

Description:	Defines the subject for notification mails sent to customers, about new password.
Group:	Framework

SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerPanelSubjectLostPassword'} = 'New OTRS password';`

2.48.20. CustomerPanelBodyLostPassword

Description:	Defines the body text for notification mails sent to customers, about new password (after using this link the new password will be sent).
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1

Config-Setting:	`$Self->{'CustomerPanelBodyLostPassword'} = 'Hi <OTRS_USERFIRSTNAME>,`
	
	`New password: <OTRS_NEWPW>`
	
	`<OTRS_CONFIG_HttpType>://<OTRS_CONFIG_FQDN>/ <OTRS_CONFIG_ScriptAlias>customer.pl ';`

2.48.21. CustomerPanelSubjectNewAccount

Description:	Defines the subject for notification mails sent to customers, about new account.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerPanelSubjectNewAccount'} = 'New OTRS Account!';`

2.48.22. CustomerPanelBodyNewAccount

Description:	Defines the body text for notification mails sent to customers, about new account.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1

Config-Setting:	`$Self->{'CustomerPanelBodyNewAccount'} = 'Hi <OTRS_USERFIRSTNAME>,`
	
	`You or someone impersonating you has created a new OTRS account for you.`
	
	`Full name: <OTRS_USERFIRSTNAME> <OTRS_USERLASTNAME>`
	`User name: <OTRS_USERLOGIN>`
	`Password : <OTRS_USERPASSWORD>`
	
	`You can log in via the following URL. We encourage you to change your password via the Preferences button after logging in.`

	<OTRS_CONFIG_HttpType>://<OTRS_CONFIG_FQDN>/ <OTRS_CONFIG_ScriptAlias>customer.pl ';

2.48.23. Loader::Customer::Skin###000-default

Description:	Default skin for OTRS 3.0 interface.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	0

Config-Setting:	$Self->{'Loader::Customer::Skin'}->{'000-default'} = {
	'Description' => 'This is the default orange - black skin for OTRS 3.0.',
	'HomePage' => 'www.otrs.org',
	'InternalName' => 'default',
	'VisibleName' => 'Default'
	};

2.48.24. Loader::Customer::SelectedSkin

Description:	The customer skin's InternalName which should be used in the customer interface. Please check the available skins in Frontend::Customer::Skins.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	$Self->{'Loader::Customer::SelectedSkin'} = 'default';

2.48.25. AutoComplete::Customer###Default

Description:	Defines the config options for the autocompletion feature.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	0

Config-Setting:	$Self->{'AutoComplete::Customer'}->{'Default'} = {
	'AutoCompleteActive' => '1',
	'ButtonText' => 'Search',
	'MaxResultsDisplayed' => '20',
	'MinQueryLength' => '2',
	'QueryDelay' => '100'
	};

2.49. Frontend::Customer::Auth

2.49.1. Customer::AuthModule

Description:	Defines the module to authenticate customers.
Group:	Framework

SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Customer::AuthModule'} = 'Kernel::System::CustomerAuth::DB';`

2.49.2. Customer::AuthModule::DB::CryptType

Description:	If "DB" was selected for Customer::AuthModule, the crypt type of passwords must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Customer::AuthModule::DB::CryptType'} = 'sha2';`

2.49.3. Customer::AuthModule::DB::Table

Description:	If "DB" was selected for Customer::AuthModule, the name of the table where your customer data should be stored must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Customer::AuthModule::DB::Table'} = 'customer_user';`

2.49.4. Customer::AuthModule::DB::CustomerKey

Description:	If "DB" was selected for Customer::AuthModule, the name of the column for the CustomerKey in the customer table must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Customer::AuthModule::DB::CustomerKey'} = 'login';`

2.49.5. Customer::AuthModule::DB::CustomerPassword

Description:	If "DB" was selected for Customer::AuthModule, the column name for the CustomerPassword in the customer table must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Customer::AuthModule::DB::CustomerPassword'} = 'pw';`

2.49.6. Customer::AuthModule::DB::DSN

Description:	If "DB" was selected for Customer::AuthModule, the DSN for the connection to the customer table must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::DB::DSN'} = 'DBI:mysql:database=customerdb;host=customerdbhost';`

2.49.7. Customer::AuthModule::DB::User

Description:	If "DB" was selected for Customer::AuthModule, a username to connect to the customer table can be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::DB::User'} = 'some_user';`

2.49.8. Customer::AuthModule::DB::Password

Description:	If "DB" was selected for Customer::AuthModule, a password to connect to the customer table can be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::DB::Password'} = 'some_password';`

2.49.9. Customer::AuthModule::DB::Type

Description:	If "DB" was selected for Customer::AuthModule, a database driver (normally autodetection is used) can be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::DB::Type'} = 'mysql';`

2.49.10. Customer::AuthModule::HTTPBasicAuth::Replace

Description:	If "HTTPBasicAuth" was selected for Customer::AuthModule, you can specify to strip leading parts of user names (e. g. for domains like example_domain\user to user).

Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::HTTPBasicAuth::Replace'} = 'example_domain\\\\';`

2.49.11. Customer::AuthModule::HTTPBasicAuth::ReplaceRegExp

Description:	If "HTTPBasicAuth" was selected for Customer::AuthModule, you can specify (by using a RegExp) to strip parts of REMOTE_USER (e. g. for to remove trailing domains). RegExp-Note, $1 will be the new Login.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::HTTPBasicAuth::ReplaceRegExp'} = '^(.+?)@.+?$';`

2.49.12. Customer::AuthModule::LDAP::Host

Description:	If "LDAP" was selected for Customer::AuthModule, the LDAP host can be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::LDAP::Host'} =  'ldap.example.com';`

2.49.13. Customer::AuthModule::LDAP::BaseDN

Description:	If "LDAP" was selected for Customer::AuthModule, the BaseDN must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::LDAP::BaseDN'} = 'dc=example,dc=com';`

2.49.14. Customer::AuthModule::LDAP::UID

Description:	If "LDAP" was selected for Customer::AuthModule, the user identifier must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth

Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::LDAP::UID'} = 'uid';`

2.49.15. Customer::AuthModule::LDAP::GroupDN

Description:	If "LDAP" was selected for Customer::Authmodule, you can check if the user is allowed to authenticate because he is in a posixGroup, e.g. user needs to be in a group xyz to use OTRS. Specify the group, who may access the system.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::LDAP::GroupDN'} = 'cn=otrsallow,ou=posixGroups,dc=example,dc=com';`

2.49.16. Customer::AuthModule::LDAP::AccessAttr

Description:	If "LDAP" was selected for Customer::AuthModule, you can specify access attributes here.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::LDAP::AccessAttr'} = 'memberUid';`

2.49.17. Customer::AuthModule::LDAP::UserAttr

Description:	If "LDAP" was selected for Customer::AuthModule, user attributes can be specified. For LDAP posixGroups use UID, for non LDAP posixGroups use full user DN.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::LDAP::UserAttr'} = 'UID';`

2.49.18. Customer::AuthModule::LDAP::SearchUserDN

Description:	If "LDAP" was selected for Customer::AuthModule and your users have only anonymous access to the LDAP tree, but you want to search through the data, you can do this with a user who has access to the LDAP directory. Specify the username for this special user here.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0

Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::LDAP::SearchUserDN'} = 'cn=binduser,ou=users,dc=example,dc=com';`

2.49.19. Customer::AuthModule::LDAP::SearchUserPw

Description:	If "LDAP" was selected for Customer::AuthModule and your users have only anonymous access to the LDAP tree, but you want to search through the data, you can do this with a user who has access to the LDAP directory. Specify the password for this special user here.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::LDAP::SearchUserPw'} = 'some_password';`

2.49.20. Customer::AuthModule::LDAP::AlwaysFilter

Description:	If "LDAP" was selected, you can add a filter to each LDAP query, e.g. (mail=*), (objectclass=user) or (!objectclass=computer).
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::LDAP::AlwaysFilter'} = '(!objectclass=computer)';`

2.49.21. Customer::AuthModule::LDAP::UserSuffix

Description:	If "LDAP" was selected for Customer::AuthModule and if you want to add a suffix to every customer login name, specifiy it here, e. g. you just want to write the username user but in your LDAP directory exists user@domain.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::LDAP::UserSuffix'} = '@domain.com';`

2.49.22. Customer::AuthModule::LDAP::Params

Description:	If "LDAP" was selected for Customer::AuthModule and special paramaters are needed for the Net::LDAP perl module, you can specify them here. See "perldoc Net::LDAP" for more information about the parameters.
Group:	Framework
SubGroup:	Frontend::Customer::Auth

Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::LDAP::Params'} = { 'async' => '0', 'port' => '389', 'timeout' => '120', 'version' => '3' };`

2.49.23. Customer::AuthModule::LDAP::Die

Description:	If "LDAP" was selected for Customer::AuthModule, you can specify if the applications will stop if e. g. a connection to a server can't be established due to network problems.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::LDAP::Die'} = '1';`

2.49.24. Customer::AuthModule::Radius::Host

Description:	If "Radius" was selected for Customer::AuthModule, the radius host must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::Radius::Host'} = 'radiushost';`

2.49.25. Customer::AuthModule::Radius::Password

Description:	If "Radius" was selected for Customer::AuthModule, the password to authenticate to the radius host must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::Radius::Password'} = 'radiussecret';`

2.49.26. Customer::AuthModule::Radius::Die

Description:	If "Radius" was selected for Customer::AuthModule, you can specify if the applications will stop if e. g. a connection to a server can't be established due to network problems.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1

Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::Radius::Die'} = '1';`

2.50. Frontend::Customer::ModuleMetaHead

2.50.1. CustomerFrontend::HeaderMetaModule###1-Refresh

Description:	Defines the module to generate html refresh headers of html sites, in the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleMetaHead
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerFrontend::HeaderMetaModule'}->{'1-Refresh'} = { 'Module' => 'Kernel::Output::HTML::HeaderMetaRefresh' };`

2.51. Frontend::Customer::ModuleNotify

2.51.1. CustomerFrontend::NotifyModule###1-ShowAgentOnline

Description:	Defines the module that shows the currently loged in agents in the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleNotify
Valid:	0
Required:	0
Config-Setting:	`$Self->{'CustomerFrontend::NotifyModule'}->{'1-ShowAgentOnline'} = { 'IdleMinutes' => '60', 'Module' => 'Kernel::Output::HTML::NotificationAgentOnline', 'ShowEmail' => '1' };`

2.51.2. CustomerFrontend::NotifyModule###1-ShowCustomerOnline

Description:	Defines the module that shows the currently loged in customers in the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleNotify
Valid:	0
Required:	0
Config-Setting:	`$Self->{'CustomerFrontend::NotifyModule'}->{'1-ShowCustomerOnline'} = { 'Module' => 'Kernel::Output::HTML::NotificationCustomerOnline', 'ShowEmail' => '1' };`

2.52. Frontend::Customer::ModuleRegistration

2.52.1. CustomerFrontend::Module###Logout

Description:	Frontend module registration for the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'CustomerFrontend::Module'}->{'Logout'} = {`
	`'Description' => 'Logout of customer panel',`
	`'NavBarName' => '',`
	`'Title' => ''`
	`};`

2.52.2. CustomerFrontend::Module###CustomerPreferences

Description:	Frontend module registration for the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'CustomerFrontend::Module'}->{'CustomerPreferences'} = {`
	`'Description' => 'Customer preferences',`
	`'NavBarName' => '',`
	`'Title' => 'Preferences'`
	`};`

2.52.3. CustomerFrontend::Module###CustomerAccept

Description:	Frontend module registration for the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'CustomerFrontend::Module'}->{'CustomerAccept'} = {`
	`'Description' => 'To accept login information, such as an EULA or license.',`
	`'NavBarName' => '',`
	`'Title' => 'Info'`
	`};`

2.52.4. CustomerFrontend::Module###PictureUpload

Description:	Frontend module registration for the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleRegistration

Valid:	1
Required:	1

Config-Setting:	`$Self->{'CustomerFrontend::Module'}->{'PictureUpload'} = {`
	`'Description' => 'Picture upload module',`
	`'NavBarName' => 'Ticket',`
	`'Title' => 'Picture-Upload'`
	`};`

2.53. Frontend::Customer::Preferences

2.53.1. PreferencesTable

Description:	Defines the name of the table, where the customer preferences are stored.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PreferencesTable'} = 'user_preferences';`

2.53.2. PreferencesTableKey

Description:	Defines the column to store the keys for the preferences table.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PreferencesTableKey'} = 'preferences_key';`

2.53.3. CustomerPreferences

Description:	Defines the parameters for the customer preferences table.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	1

Config-Setting:	`$Self->{'CustomerPreferences'} = {`
	`'Module' => 'Kernel::System::CustomerUser::Preferences::DB',`
	`'Params' => {`
	`'Table' => 'customer_preferences',`
	`'TableKey' => 'preferences_key',`
	`'TableUserID' => 'user_id',`
	`'TableValue' => 'preferences_value'`
	`}`
	`};`

2.53.4. CustomerPreferencesView

Description:	Sets the order of the different items in the customer preferences view.

Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerPreferencesView'} = [ 'User Profile', 'Other Settings' ];`

2.53.5. CustomerPreferencesGroups###Password

Description:	Defines all the parameters for this item in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerPreferencesGroups'}->{'Password'} = { 'Active' => '1', 'Area' => 'Customer', 'Column' => 'Other Settings', 'Label' => 'Change password', 'Module' => 'Kernel::Output::HTML::PreferencesPassword', 'PasswordMin2Characters' => '0', 'PasswordMin2Lower2UpperCharacters' => '0', 'PasswordMinSize' => '0', 'PasswordNeedDigit' => '0', 'PasswordRegExp' => '', 'Prio' => '1000' };`

2.53.6. CustomerPreferencesGroups###Language

Description:	Defines all the parameters for this item in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerPreferencesGroups'}->{'Language'} = { 'Active' => '1', 'Column' => 'User Profile', 'Key' => 'Language', 'Label' => 'Interface language', 'Module' => 'Kernel::Output::HTML::PreferencesLanguage', 'PrefKey' => 'UserLanguage', 'Prio' => '2000' };`

2.53.7. CustomerPreferencesGroups###Theme

Description:	Defines all the parameters for this item in the customer preferences.
Group:	Framework

SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0

Config-Setting:	`$Self->{'CustomerPreferencesGroups'}->{'Theme'} = {`
	`'Active' => '0',`
	`'Column' => 'User Profile',`
	`'Key' => 'Select your frontend Theme.',`
	`'Label' => 'Theme',`
	`'Module' => 'Kernel::Output::HTML::PreferencesTheme',`
	`'PrefKey' => 'UserTheme',`
	`'Prio' => '1000'`
	`};`

2.53.8. CustomerPreferencesGroups###TimeZone

Description:	Defines all the parameters for this item in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0

Config-Setting:	`$Self->{'CustomerPreferencesGroups'}->{'TimeZone'} = {`
	`'Active' => '1',`
	`'Column' => 'User Profile',`
	`'Key' => 'Time Zone',`
	`'Label' => 'Time Zone',`
	`'Module' => 'Kernel::Output::HTML::PreferencesTimeZone',`
	`'PrefKey' => 'UserTimeZone',`
	`'Prio' => '5000'`
	`};`

2.53.9. CustomerPreferencesGroups###PGP

Description:	Defines all the parameters for this item in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0

Config-Setting:	`$Self->{'CustomerPreferencesGroups'}->{'PGP'} = {`
	`'Active' => '1',`
	`'Column' => 'Other Settings',`
	`'Key' => 'PGP Key Upload',`
	`'Label' => 'PGP Key',`
	`'Module' => 'Kernel::Output::HTML::PreferencesPGP',`
	`'PrefKey' => 'UserPGPKey',`
	`'Prio' => '10000'`
	`};`

2.53.10. CustomerPreferencesGroups###SMIME

Description:	Defines all the parameters for this item in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences

Valid:	1
Required:	0

Config-Setting:	`$Self->{'CustomerPreferencesGroups'}->{'SMIME'} = {`
	`'Active' => '1',`
	`'Column' => 'Other Settings',`
	`'Key' => 'S/MIME Certificate Upload',`
	`'Label' => 'S/MIME Certificate',`
	`'Module' => 'Kernel::Output::HTML::PreferencesSMIME',`
	`'PrefKey' => 'UserSMIMEKey',`
	`'Prio' => '11000'`
	`};`

2.54. Frontend::Public

2.54.1. PublicFrontend::CommonParam###Action

Description:	Defines the default value for the action parameter for the public frontend. The action parameter is used in the scripts of the system.
Group:	Framework
SubGroup:	Frontend::Public
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PublicFrontend::CommonParam'}->{'Action'} = 'PublicDefault';`

2.55. Frontend::Public::ModuleRegistration

2.55.1. PublicFrontend::Module###PublicDefault

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

Config-Setting:	`$Self->{'PublicFrontend::Module'}->{'PublicDefault'} = {`
	`'Description' => 'PublicDefault',`
	`'NavBarName' => '',`
	`'Title' => 'PublicDefault'`
	`};`

2.55.2. PublicFrontend::Module###PublicRepository

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

Config-Setting:	`$Self->{'PublicFrontend::Module'}->{'PublicRepository'} = {`
	`'Description' => 'PublicRepository',`
	`'NavBarName' => '',`
	`'Title' => 'PublicRepository'`
	`};`

2.55.3. PublicFrontend::Module###PublicSupportDataCollector

Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Public::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'PublicFrontend::Module'}->{'PublicSupportDataCollector'} = { 'Description' => 'PublicSupportDataCollector', 'NavBarName' => '', 'Title' => 'PublicSupportDataCollector' };`

3. GenericInterface

3.1. Core::CustomerCompany

3.1.1. CustomerCompany::EventModulePost###1000-GenericInterface

Description:	Performs the configured action for each event (as an Invoker) for each configured Webservice.
Group:	GenericInterface
SubGroup:	Core::CustomerCompany
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerCompany::EventModulePost'}->{'1000-GenericInterface'} = { 'Event' => '', 'Module' => 'Kernel::GenericInterface::Event::Handler', 'Transaction' => '1' };`

3.2. Core::CustomerUser

3.2.1. CustomerUser::EventModulePost###1000-GenericInterface

Description:	Performs the configured action for each event (as an Invoker) for each configured Webservice.
Group:	GenericInterface
SubGroup:	Core::CustomerUser
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerUser::EventModulePost'}->{'1000-GenericInterface'} = { 'Event' => '',`

	<pre> 'Module' => 'Kernel::GenericInterface::Event::Handler', 'Transaction' => '1' };</pre>

3.3. Core::Package

3.3.1. Package::EventModulePost###1000-GenericInterface

Description:	Performs the configured action for each event (as an Invoker) for each configured Webservice.
Group:	GenericInterface
SubGroup:	Core::Package
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Package::EventModulePost'}->{'1000-GenericInterface'} = {
	  'Event' => '',
	  'Module' => 'Kernel::GenericInterface::Event::Handler',
	  'Transaction' => '1'
	};</pre>

3.4. Core::Ticket

3.4.1. Ticket::EventModulePost###1000-GenericInterface

Description:	Performs the configured action for each event (as an Invoker) for each configured Webservice.
Group:	GenericInterface
SubGroup:	Core::Ticket
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Ticket::EventModulePost'}->{'1000-GenericInterface'} = {
	  'Event' => '',
	  'Module' => 'Kernel::GenericInterface::Event::Handler',
	  'Transaction' => '1'
	};</pre>

3.5. Frontend::Admin::ModuleRegistration

3.5.1. Frontend::Module###AdminGenericInterfaceDebugger

Description:	Frontend module registration for the agent interface.
Group:	GenericInterface
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AdminGenericInterfaceDebugger'} = {</pre>

```
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.GenericInterface.css'
    ],
    'CSS_IE7' => 'Core.Agent.Admin.GenericInterface.IE7.css',
    'JavaScript' => [
      'Core.Agent.Admin.GenericInterfaceDebugger.js'
    ]
  },
  'Title' => 'GenericInterface Debugger GUI'
};
```

3.5.2. Frontend::Module###AdminGenericInterfaceWebservice

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

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

3.5.3. Frontend::Module###AdminGenericInterfaceTransportHTTPSOA

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

```
 $Self->{'Frontend::Module'}-
>{'AdminGenericInterfaceTransportHTTPSOAP'} =  {
  'Description' => 'Admin',
```

```
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.GenericInterface.css'
    ]
  },
  'Title' => 'GenericInterface TransportHTTPSOAP GUI'
};
```

3.5.4. Frontend::Module###AdminGenericInterfaceWebserviceHistory

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

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

3.5.5. Frontend::Module###AdminGenericInterfaceOperationDefault

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

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

3.5.6. Frontend::Module###AdminGenericInterfaceInvokerDefault

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

Config-Setting:	<pre>$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'
	};</pre>

3.5.7. Frontend::Module###AdminGenericInterfaceMappingSimple

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

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

3.6. GenericInterface::Invoker::ModuleRegistration

3.6.1. GenericInterface::Invoker::Module###Test::Test

Description:	GenericInterface module registration for the invoker layer.

Group:	GenericInterface
SubGroup:	GenericInterface::Invoker::ModuleRegistration
Valid:	0
Required:	0

Config-Setting:	`$Self->{'GenericInterface::Invoker::Module'}->{'Test::Test'} = {`
	`  'ConfigDialog' => 'AdminGenericInterfaceInvokerDefault',`
	`  'Controller' => 'Test',`
	`  'Name' => 'Test'`
	`};`

3.6.2. GenericInterface::Invoker::Module###Test::TestSimple

Description:	GenericInterface module registration for the invoker layer.
Group:	GenericInterface
SubGroup:	GenericInterface::Invoker::ModuleRegistration
Valid:	0
Required:	0

Config-Setting:	`$Self->{'GenericInterface::Invoker::Module'}->{'Test::TestSimple'}`
	`= {`
	`  'ConfigDialog' => 'AdminGenericInterfaceInvokerDefault',`
	`  'Controller' => 'Test',`
	`  'Name' => 'TestSimple'`
	`};`

3.7. GenericInterface::Mapping::ModuleRegistration

3.7.1. GenericInterface::Mapping::Module###Test

Description:	GenericInterface module registration for the mapping layer.
Group:	GenericInterface
SubGroup:	GenericInterface::Mapping::ModuleRegistration
Valid:	0
Required:	0

Config-Setting:	`$Self->{'GenericInterface::Mapping::Module'}->{'Test'} = {`
	`  'ConfigDialog' => ''`
	`};`

3.7.2. GenericInterface::Mapping::Module###Simple

Description:	GenericInterface module registration for the mapping layer.
Group:	GenericInterface
SubGroup:	GenericInterface::Mapping::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'GenericInterface::Mapping::Module'}->{'Simple'} = {`
	`  'ConfigDialog' => 'AdminGenericInterfaceMappingSimple'`
	`};`

3.8. GenericInterface::Operation::ModuleRegistration

3.8.1. GenericInterface::Operation::Module###Test::Test

Description:	GenericInterface module registration for the operation layer.
Group:	GenericInterface
SubGroup:	GenericInterface::Operation::ModuleRegistration
Valid:	0
Required:	0
Config-Setting:	`$Self->{'GenericInterface::Operation::Module'}->{'Test::Test'} = { 'ConfigDialog' => 'AdminGenericInterfaceOperationDefault', 'Controller' => 'Test', 'Name' => 'Test' };`

3.8.2. GenericInterface::Operation::Module###Session::SessionCreate

Description:	GenericInterface module registration for the operation layer.
Group:	GenericInterface
SubGroup:	GenericInterface::Operation::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'GenericInterface::Operation::Module'}->{'Session::SessionCreate'} = { 'ConfigDialog' => 'AdminGenericInterfaceOperationDefault', 'Controller' => 'Session', 'Name' => 'SessionCreate' };`

3.8.3. GenericInterface::Operation::Module###Ticket::TicketCreate

Description:	GenericInterface module registration for the operation layer.
Group:	GenericInterface
SubGroup:	GenericInterface::Operation::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'GenericInterface::Operation::Module'}->{'Ticket::TicketCreate'} = { 'ConfigDialog' => 'AdminGenericInterfaceOperationDefault', 'Controller' => 'Ticket', 'Name' => 'TicketCreate' };`

3.8.4. GenericInterface::Operation::Module###Ticket::TicketUpdate

Description:	GenericInterface module registration for the operation layer.

Group:	GenericInterface
SubGroup:	GenericInterface::Operation::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'GenericInterface::Operation::Module'}->{'Ticket::TicketUpdate'} = {
	'ConfigDialog' => 'AdminGenericInterfaceOperationDefault',
	'Controller' => 'Ticket',
	'Name' => 'TicketUpdate'
	};</pre>

3.8.5. GenericInterface::Operation::Module###Ticket::TicketGet

Description:	GenericInterface module registration for the operation layer.
Group:	GenericInterface
SubGroup:	GenericInterface::Operation::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'GenericInterface::Operation::Module'}->{'Ticket::TicketGet'} = {
	'ConfigDialog' => 'AdminGenericInterfaceOperationDefault',
	'Controller' => 'Ticket',
	'Name' => 'TicketGet'
	};</pre>

3.8.6. GenericInterface::Operation::Module###Ticket::TicketSearch

Description:	GenericInterface module registration for the operation layer.
Group:	GenericInterface
SubGroup:	GenericInterface::Operation::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'GenericInterface::Operation::Module'}->{'Ticket::TicketSearch'} = {
	'ConfigDialog' => 'AdminGenericInterfaceOperationDefault',
	'Controller' => 'Ticket',
	'Name' => 'TicketGet'
	};</pre>

3.9. GenericInterface::Operation::TicketCreate

3.9.1. GenericInterface::Operation::TicketCreate###ArticleType

Description:	Defines the default type of the article for this operation.
Group:	GenericInterface
SubGroup:	GenericInterface::Operation::TicketCreate
Valid:	1

Required:	1
Config-Setting:	`$Self->{'GenericInterface::Operation::TicketCreate'}->{'ArticleType'} = 'webrequest';`

3.9.2. GenericInterface::Operation::TicketCreate###HistoryType

Description:	Defines the history type for this operation, which gets used for ticket history in the agent interface.
Group:	GenericInterface
SubGroup:	GenericInterface::Operation::TicketCreate
Valid:	1
Required:	1
Config-Setting:	`$Self->{'GenericInterface::Operation::TicketCreate'}->{'HistoryType'} = 'NewTicket';`

3.9.3. GenericInterface::Operation::TicketCreate###HistoryComment

Description:	Defines the history comment for this operation, which gets used for ticket history in the agent interface.
Group:	GenericInterface
SubGroup:	GenericInterface::Operation::TicketCreate
Valid:	1
Required:	1
Config-Setting:	`$Self->{'GenericInterface::Operation::TicketCreate'}->{'HistoryComment'} = '%%GenericInterface Create';`

3.9.4. GenericInterface::Operation::TicketCreate###AutoResponseTyp

Description:	Defines the default auto response type of the article for this operation.
Group:	GenericInterface
SubGroup:	GenericInterface::Operation::TicketCreate
Valid:	1
Required:	1
Config-Setting:	`$Self->{'GenericInterface::Operation::TicketCreate'}->{'AutoResponseType'} = 'auto reply';`

3.10. GenericInterface::Operation::TicketSearch

3.10.1. GenericInterface::Operation::TicketSearch###SearchLimit

Description:	Maximum number of tickets to be displayed in the result of this operation.
Group:	GenericInterface

SubGroup:	GenericInterface::Operation::TicketSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'GenericInterface::Operation::TicketSearch'}->{'SearchLimit'} = '500';`

3.10.2. GenericInterface::Operation::TicketSearch###SortBy::Default

Description:	Defines the default ticket attribute for ticket sorting of the ticket search result of this operation.
Group:	GenericInterface
SubGroup:	GenericInterface::Operation::TicketSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'GenericInterface::Operation::TicketSearch'}->{'SortBy::Default'} = 'Age';`

3.10.3. GenericInterface::Operation::TicketSearch###Order::Default

Description:	Defines the default ticket order in the ticket search result of the this operation. Up: oldest on top. Down: latest on top.
Group:	GenericInterface
SubGroup:	GenericInterface::Operation::TicketSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'GenericInterface::Operation::TicketSearch'}->{'Order::Default'} = 'Down';`

3.11. GenericInterface::Operation::TicketUpdate

3.11.1. GenericInterface::Operation::TicketUpdate###ArticleType

Description:	Defines the default type of the article for this operation.
Group:	GenericInterface
SubGroup:	GenericInterface::Operation::TicketUpdate
Valid:	1
Required:	1
Config-Setting:	`$Self->{'GenericInterface::Operation::TicketUpdate'}->{'ArticleType'} = 'webrequest';`

3.11.2. GenericInterface::Operation::TicketUpdate###HistoryType

Description:	Defines the history type for this operation, which gets used for ticket history in the agent interface.

Group:	GenericInterface
SubGroup:	GenericInterface::Operation::TicketUpdate
Valid:	1
Required:	1
Config-Setting:	`$Self->{'GenericInterface::Operation::TicketUpdate'}->{'HistoryType'} = 'AddNote';`

3.11.3. GenericInterface::Operation::TicketUpdate###HistoryComment

Description:	Defines the history comment for this operation, which gets used for ticket history in the agent interface.
Group:	GenericInterface
SubGroup:	GenericInterface::Operation::TicketUpdate
Valid:	1
Required:	1
Config-Setting:	`$Self->{'GenericInterface::Operation::TicketUpdate'}->{'HistoryComment'} = '%%GenericInterface Note';`

3.11.4. GenericInterface::Operation::TicketUpdate###AutoResponseTyp

Description:	Defines the default auto response type of the article for this operation.
Group:	GenericInterface
SubGroup:	GenericInterface::Operation::TicketUpdate
Valid:	1
Required:	1
Config-Setting:	`$Self->{'GenericInterface::Operation::TicketUpdate'}->{'AutoResponseType'} = 'auto follow up';`

3.12. GenericInterface::Transport::ModuleRegistration

3.12.1. GenericInterface::Transport::Module###HTTP::SOAP

Description:	GenericInterface module registration for the transport layer.
Group:	GenericInterface
SubGroup:	GenericInterface::Transport::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'GenericInterface::Transport::Module'}->{'HTTP::SOAP'} = { 'ConfigDialog' => 'AdminGenericInterfaceTransportHTTPSOAP', 'Name' => 'SOAP', 'Protocol' => 'HTTP' };`

3.12.2. GenericInterface::Transport::Module###HTTP::Test

Description:	GenericInterface module registration for the transport layer.
Group:	GenericInterface
SubGroup:	GenericInterface::Transport::ModuleRegistration
Valid:	0
Required:	0
Config-Setting:	`$Self->{'GenericInterface::Transport::Module'}->{'HTTP::Test'} = { 'ConfigDialog' => 'AdminGenericInterfaceTransportHTTPTest', 'Name' => 'Test', 'Protocol' => 'HTTP' };`

3.13. GenericInterface::Webservice

3.13.1. GenericInterface::WebserviceConfig::CacheTTL

Description:	Cache time in seconds for the web service config backend.
Group:	GenericInterface
SubGroup:	GenericInterface::Webservice
Valid:	1
Required:	1
Config-Setting:	`$Self->{'GenericInterface::WebserviceConfig::CacheTTL'} = '86400';`

3.13.2. GenericInterface::Operation::Common::CachedAuth::AgentCache

Description:	Cache time in seconds for agent authentication in the GenericInterface.
Group:	GenericInterface
SubGroup:	GenericInterface::Webservice
Valid:	1
Required:	1
Config-Setting:	`$Self->{'GenericInterface::Operation::Common::CachedAuth::AgentCacheTTL'} = '300';`

3.13.3. GenericInterface::Operation::Common::CachedAuth::CustomerCa

Description:	Cache time in seconds for customer authentication in the GenericInterface.
Group:	GenericInterface
SubGroup:	GenericInterface::Webservice
Valid:	1
Required:	1

Config-Setting:	`$Self->{'GenericInterface::Operation::Common::CachedAuth::CustomerCacheTTL'} = '300';`

3.13.4. GenericInterface::Webservice::Path::Separator

Description:	Webservice path separator.
Group:	GenericInterface
SubGroup:	GenericInterface::Webservice
Valid:	1
Required:	1
Config-Setting:	`$Self->{'GenericInterface::Webservice::Path::Separator'} = '»';`

4. ProcessManagement

4.1. Core

4.1.1. Process::DynamicFieldProcessManagementProcessID

Description:	This option defines the dynamic field in which a Process Management process entity id is stored.
Group:	ProcessManagement
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Process::DynamicFieldProcessManagementProcessID'} = 'ProcessManagementProcessID';`

4.1.2. Process::DynamicFieldProcessManagementActivityID

Description:	This option defines the dynamic field in which a Process Management activity entity id is stored.
Group:	ProcessManagement
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Process::DynamicFieldProcessManagementActivityID'} = 'ProcessManagementActivityID';`

4.1.3. Process::DefaultQueue

Description:	This option defines the process tickets default queue.
Group:	ProcessManagement
SubGroup:	Core
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Process::DefaultQueue'} = 'Raw';`

4.1.4. Process::DefaultState

Description:	This option defines the process tickets default state.
Group:	ProcessManagement
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Process::DefaultState'} = 'new';`

4.1.5. Process::DefaultLock

Description:	This option defines the process tickets default lock.
Group:	ProcessManagement
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Process::DefaultLock'} = 'unlock';`

4.1.6. Process::DefaultPriority

Description:	This option defines the process tickets default priority.
Group:	ProcessManagement
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Process::DefaultPriority'} = '3 normal';`

4.1.7. Process::Entity::Prefix

Description:	Default ProcessManagement entity prefixes for entity IDs that are automatically generated.
Group:	ProcessManagement
SubGroup:	Core
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Process::Entity::Prefix'} = {`
	`'Activity' => 'A',`
	`'ActivityDialog' => 'AD',`
	`'Process' => 'P',`
	`'Transition' => 'T',`
	`'TransitionAction' => 'TA'`
	`};`

4.1.8. Process::CacheTTL

Description:	Cache time in seconds for the DB process backend.
Group:	ProcessManagement
SubGroup:	Core

Valid:	1
Required:	1
Config-Setting:	`$Self->{'Process::CacheTTL'} = '3600';`

4.1.9. Process::NavBarOutput::CacheTTL

Description:	Cache time in seconds for the ticket process navigation bar output module.
Group:	ProcessManagement
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Process::NavBarOutput::CacheTTL'} = '900';`

4.2. Core::Ticket

4.2.1. Ticket::EventModulePost###TicketProcessTransitions

Description:	Event module registration. For more performance you can define a trigger event (e. g. Event => TicketCreate).
Group:	ProcessManagement
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'TicketProcessTransitions'} = { 'Event' => '', 'Module' => 'Kernel::System::Ticket::Event::TicketProcessTransitions', 'Transaction' => '1' };`

4.3. Frontend::Admin::ModuleRegistration

4.3.1. Frontend::Module###AdminProcessManagement

Description:	Frontend module registration for the agent interface.
Group:	ProcessManagement
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminProcessManagement'} = { 'Description' => 'Admin', 'Group' => [ 'admin' ], 'Loader' => { 'CSS' => [ 'Core.Agent.Admin.ProcessManagement.css', 'Core.AllocationList.css'`

```
    ],
    'JavaScript' => [
      'thirdparty/jsplumb-1.5.5/jsplumb.js',
      'thirdparty/farahey-0.1/farahey.js',
      'thirdparty/jsplumb-labelspacer/label-spacer.js',
      'Core.Agent.Admin.ProcessManagement.js',
      'Core.Agent.Admin.ProcessManagement.Canvas.js',
      'Core.UI.AllocationList.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'System',
    'Description' => 'Configure Processes.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Process Management',
    'Prio' => '750'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Process Management'
};
```

4.3.2. Frontend::Module###AdminProcessManagementActivity

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

```
 $Self->{'Frontend::Module'}->{'AdminProcessManagementActivity'} =
 {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.ProcessManagement.css',
      'Core.AllocationList.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.ProcessManagement.js',
      'Core.UI.AllocationList.js'
    ]
  },
  'Title' => 'Process Management Activity GUI'
};
```

4.3.3. Frontend::Module###AdminProcessManagementActivityDialog

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

```
 $Self->{'Frontend::Module'}-
>{'AdminProcessManagementActivityDialog'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
```

```
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.ProcessManagement.css',
      'Core.AllocationList.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.ProcessManagement.js',
      'Core.UI.AllocationList.js'
    ]
  },
  'Title' => 'Process Management Activity Dialog GUI'
};
```

4.3.4. Frontend::Module###AdminProcessManagementTransition

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

```
 $Self->{'Frontend::Module'}->{'AdminProcessManagementTransition'} =
  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.ProcessManagement.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.ProcessManagement.js'
    ]
  },
  'Title' => 'Process Management Transition GUI'
};
```

4.3.5. Frontend::Module###AdminProcessManagementTransitionActio

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

```
 $Self->{'Frontend::Module'}-
>{'AdminProcessManagementTransitionAction'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.ProcessManagement.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.ProcessManagement.js'
    ]
  },
```

```
  'Title' => 'Process Management Transition Action GUI'
};
```

4.3.6. Frontend::Module###AdminProcessManagementPath

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

```
$Self->{'Frontend::Module'}->{'AdminProcessManagementPath'} =  {
 'Description' => 'Admin',
 'Group' => [
   'admin'
 ],
 'Loader' => {
   'CSS' => [
     'Core.Agent.Admin.ProcessManagement.css',
     'Core.AllocationList.css'
   ],
   'JavaScript' => [
     'Core.Agent.Admin.ProcessManagement.js',
     'Core.UI.AllocationList.js'
   ]
 },
 'Title' => 'Process Management Path GUI'
};
```

4.4. Frontend::Agent::ModuleRegistration

4.4.1. Frontend::Module###AgentTicketProcess

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

```
$Self->{'Frontend::Module'}->{'AgentTicketProcess'} =  {
 'Description' => 'Create new process ticket',
 'Loader' => {
   'CSS' => [
     'Core.Agent.TicketProcess.css'
   ],
   'JavaScript' => [
     'Core.Agent.CustomerSearch.js',
     'Core.Agent.TicketAction.js'
   ]
 },
 'NavBar' => [
   {
     'AccessKey' => 'p',
     'Block' => '',
     'Description' => 'Create New process ticket',
     'Link' => 'Action=AgentTicketProcess',
     'LinkOption' => '',
     'Name' => 'New process ticket',
     'NavBar' => 'Ticket',
     'Prio' => '220',
     'Type' => ''
   }
```

	], 'NavBarName' => 'Ticket', 'Title' => 'New process ticket' };

4.5. Frontend::Agent::NavBarModule

4.5.1. Frontend::NavBarOutputModule###1-TicketProcesses

Description:	Frontend module registration (disable ticket processes screen if no process available).
Group:	ProcessManagement
SubGroup:	Frontend::Agent::NavBarModule
Valid:	1
Required:	0

Config-Setting:	$Self->{'Frontend::NavBarOutputModule'}->{'1-TicketProcesses'} = {
	'Module' =>
	'Kernel::Output::HTML::NavBarOutputModuleAgentTicketProcess'
	};

4.6. Frontend::Agent::Ticket::ViewProcess

4.6.1. Ticket::Frontend::AgentTicketProcess###StateType

Description:	Determines the next possible ticket states, for process tickets in the agent interface.
Group:	ProcessManagement
SubGroup:	Frontend::Agent::Ticket::ViewProcess
Valid:	1
Required:	1
Config-Setting:	$Self->{'Ticket::Frontend::AgentTicketProcess'}->{'StateType'} = ['new', 'open', 'pending auto', 'pending reminder', 'closed'];

4.6.2. Ticket::Frontend::CustomerTicketProcess###StateType

Description:	Determines the next possible ticket states, for process tickets in the agent interface.
Group:	ProcessManagement
SubGroup:	Frontend::Agent::Ticket::ViewProcess
Valid:	1
Required:	1
Config-Setting:	$Self->{'Ticket::Frontend::CustomerTicketProcess'}->{'StateType'} = ['new', 'open'

	];

4.7. Frontend::Agent::Ticket::ViewZoom

4.7.1. Ticket::Frontend::AgentTicketZoom###ProcessDisplay

Description:	Display settings to override defaults for Process Tickets.
Group:	ProcessManagement
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketZoom'}->{'ProcessDisplay'} = { 'NavBarName' => 'Processes', 'WidgetTitle' => 'Process Information' };`

4.7.2. Ticket::Frontend::AgentTicketZoom###ProcessWidgetDynamicF

Description:	Dynamic fields groups for process widget. The key is the name of the group, the value contains the fields to be shown. Example: 'Key => My Group', 'Content: Name_X, NameY'.
Group:	ProcessManagement
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketZoom'}->{'ProcessWidgetDynamicFieldGroups'} = {};`

4.7.3. Ticket::Frontend::AgentTicketZoom###ProcessWidgetDynamicF

Description:	Dynamic fields shown in the process widget in ticket zoom screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled.
Group:	ProcessManagement
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketZoom'}->{'ProcessWidgetDynamicField'} = {};`

4.8. Frontend::Customer::ModuleRegistration

4.8.1. CustomerFrontend::Module###CustomerTicketProcess

Description:	Frontend module registration for the customer interface.

Group:	ProcessManagement
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'CustomerFrontend::Module'}->{'CustomerTicketProcess'} = {
	'Description' => 'Process Ticket',
	'Loader' => {
	'CSS' => [
	'Core.Customer.TicketProcess.css'
	]
	},
	'NavBar' => [
	{
	'AccessKey' => 'o',
	'Block' => '',
	'Description' => 'Create new process ticket',
	'Link' => 'Action=CustomerTicketProcess',
	'LinkOption' => '',
	'Name' => 'New process ticket',
	'NavBar' => 'Ticket',
	'Prio' => '220',
	'Type' => 'Submenu'
	}
	],
	'NavBarName' => 'Ticket',
	'Title' => 'Process ticket'
	};</pre>

4.9. Frontend::Customer::NavBarModule

4.9.1. CustomerFrontend::NavBarModule###10-CustomerTicketProcesses

Description:	Frontend module registration (disable ticket processes screen if no process available) for Customer.
Group:	ProcessManagement
SubGroup:	Frontend::Customer::NavBarModule
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'CustomerFrontend::NavBarModule'}->{'10-CustomerTicketProcesses'} = {
	'Module' =>
	'Kernel::Output::HTML::NavBarOutputModuleCustomerTicketProcess'
	};</pre>

5. Scheduler

5.1. Core

5.1.1. Scheduler::SleepTime

Description:	Defines scheduler sleep time in seconds after processing all available tasks (floating point number).
Group:	Scheduler
SubGroup:	Core

Valid:	1
Required:	1
Config-Setting:	`$Self->{'Scheduler::SleepTime'} = '1.0';`

5.1.2. Scheduler::PIDUpdateTime

Description:	Defines scheduler PID update time in seconds.
Group:	Scheduler
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Scheduler::PIDUpdateTime'} = '600';`

5.1.3. Scheduler::RestartAfterSeconds

Description:	Defines the time in seconds after which the Scheduler performs an automatic self-restart.
Group:	Scheduler
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Scheduler::RestartAfterSeconds'} = '86400';`

5.1.4. Scheduler::TaskDataLength

Description:	Defines the maximum length (in characters) for a scheduler task data. WARNING: Do not modify this setting unless you are sure of the current Database length for 'task_data' filed from 'scheduler_data_list' table.
Group:	Scheduler
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Scheduler::TaskDataLength'} = '8000';`

5.2. Core::Log

5.2.1. Scheduler::LogPath

Description:	Defines the path for scheduler to store its console output (SchedulerOUT.log and SchedulerERR.log).
Group:	Scheduler
SubGroup:	Core::Log
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Scheduler::LogPath'} = '<OTRS_CONFIG_Home>/var/log';`

5.2.2. Scheduler::Log::DaysToKeep

Description:	Defines the time in days to keep log backup files.
Group:	Scheduler
SubGroup:	Core::Log
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Scheduler::Log::DaysToKeep'} =  '10';`

5.3. Core::Web

5.3.1. Loader::Agent::CommonJS###000-GenericInterface

Description:	List of JS files to always be loaded for the agent interface.
Group:	Scheduler
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Loader::Agent::CommonJS'}->{'000-GenericInterface'} =  [ 'Core.Agent.Admin.Scheduler.js' ];`

5.4. Frontend::Admin::ModuleRegistration

5.4.1. Frontend::Module###AdminScheduler

Description:	Frontend module registration for the agent interface.
Group:	Scheduler
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminScheduler'} =  { 'Description' => 'Admin', 'Group' => [ 'admin' ], 'Title' => 'GenericInterface Webservice Mapping GUI' };`

5.5. Frontend::Agent::ModuleNotify

5.5.1. Frontend::NotifyModule###800-Scheduler-Check

Description:	Defines the module to display a notification in the agent interface if the scheduler is not running.
Group:	Scheduler
SubGroup:	Frontend::Agent::ModuleNotify

Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::NotifyModule'}->{'800-Scheduler-Check'} = { 'Module' => 'Kernel::Output::HTML::NotificationSchedulerCheck' };`

6. Ticket

6.1. Core

6.1.1. OTRSEscalationEvents::DecayTime

Description:	The duration in minutes after emitting an event, in which the new escalation notify and start events are suppressed.
Group:	Ticket
SubGroup:	Core
Valid:	1
Required:	0
Config-Setting:	`$Self->{'OTRSEscalationEvents::DecayTime'} = '1440';`

6.2. Core::CustomerCompany

6.2.1. CustomerCompany::EventModulePost###110-UpdateTickets

Description:	Event module that updates tickets after an update of the Customer Company.
Group:	Ticket
SubGroup:	Core::CustomerCompany
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerCompany::EventModulePost'}->{'110-UpdateTickets'} = { 'Event' => 'CustomerCompanyUpdate', 'Module' => 'Kernel::System::CustomerCompany::Event::TicketUpdate', 'Transaction' => '0' };`

6.2.2. CustomerUser::EventModulePost###120-UpdateTickets

Description:	Event module that updates tickets after an update of the Customer User.
Group:	Ticket
SubGroup:	Core::CustomerCompany
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerUser::EventModulePost'}->{'120-UpdateTickets'} = {`

	`'Event' => 'CustomerUserUpdate',`
	`'Module' => 'Kernel::System::CustomerUser::Event::TicketUpdate',`
	`'Transaction' => '0'`
	`};`

6.3. Core::FulltextSearch

6.3.1. Ticket::SearchIndexModule

Description:	Helps to extend your articles full-text search (From, To, Cc, Subject and Body search). Runtime will do full-text searches on live data (it works fine for up to 50.000 tickets). StaticDB will strip all articles and will build an index after article creation, increasing fulltext searches about 50%. To create an initial index use "bin/otrs.RebuildFulltextIndex.pl".
Group:	Ticket
SubGroup:	Core::FulltextSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::SearchIndexModule'} =` `'Kernel::System::Ticket::ArticleSearchIndex::RuntimeDB';`

6.3.2. Ticket::SearchIndex::Attribute

Description:	Basic fulltext index settings. Execute "bin/otrs.RebuildFulltextIndex.pl" in order to generate a new index.
Group:	Ticket
SubGroup:	Core::FulltextSearch
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::SearchIndex::Attribute'} = {`
	`'WordCountMax' => '1000',`
	`'WordLengthMax' => '30',`
	`'WordLengthMin' => '3'`
	`};`

6.3.3. Ticket::SearchIndex::Filters

Description:	Fulltext index regex filters to remove parts of the text.
Group:	Ticket
SubGroup:	Core::FulltextSearch
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::SearchIndex::Filters'} = [`
	`'[,\\&\\<\\>\\?"\\!\\*\\|;\\[\\]\\(\\)\\+\\$\\^=]',`
	`'^[\':.]|[\':.]$'`
	`];`

6.3.4. Ticket::SearchIndex::StopWords

Description:	Stop words for fulltext index. These words will be removed.
Group:	Ticket

SubGroup:	Core::FulltextSearch
Valid:	1
Required:	1

Config-Setting:	<pre>$Self->{'Ticket::SearchIndex::StopWords'} = {
	'a' => '1',
	'als' => '1',
	'and' => '1',
	'auf' => '1',
	'das' => '1',
	'den' => '1',
	'der' => '1',
	'die' => '1',
	'for' => '1',
	'im' => '1',
	'in' => '1',
	'is' => '1',
	'of' => '1',
	'the' => '1',
	'to' => '1',
	'und' => '1',
	'vom' => '1',
	'zu' => '1'
	};</pre>

6.3.5. Ticket::EventModulePost###98-ArticleSearchIndex

Description:	Builds an article index right after the article's creation.
Group:	Ticket
SubGroup:	Core::FulltextSearch
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Ticket::EventModulePost'}->{'98-ArticleSearchIndex'} = {
	'Event' => '(ArticleCreate|ArticleUpdate)',
	'Module' => 'Kernel::System::Ticket::Event::ArticleSearchIndex'
	};</pre>

6.4. Core::LinkObject

6.4.1. LinkObject::PossibleLink###0200

Description:	Links 2 tickets with a "Normal" type link.
Group:	Ticket
SubGroup:	Core::LinkObject
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'LinkObject::PossibleLink'}->{'0200'} = {
	'Object1' => 'Ticket',
	'Object2' => 'Ticket',
	'Type' => 'Normal'
	};</pre>

6.4.2. LinkObject::PossibleLink###0201

Description:	Links 2 tickets with a "ParentChild" type link.
Group:	Ticket

SubGroup:	Core::LinkObject
Valid:	1
Required:	0

Config-Setting:	`$Self->{'LinkObject::PossibleLink'}->{'0201'} = {`
	`'Object1' => 'Ticket',`
	`'Object2' => 'Ticket',`
	`'Type' => 'ParentChild'`
	`};`

6.5. Core::PostMaster

6.5.1. PostmasterMaxEmails

Description:	Maximal auto email responses to own email-address a day (Loop-Protection).
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterMaxEmails'} = '40';`

6.5.2. PostMasterMaxEmailSize

Description:	Maximal size in KBytes for mails that can be fetched via POP3/POP3S/IMAP/IMAPS (KBytes).
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostMasterMaxEmailSize'} = '16384';`

6.5.3. PostMasterReconnectMessage

Description:	The "bin/PostMasterMailAccount.pl" will reconnect to POP3/POP3S/IMAP/IMAPS host after the specified count of messages.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostMasterReconnectMessage'} = '20';`

6.5.4. LoopProtectionModule

Description:	Default loop protection module.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1

Config-Setting:	`$Self->{'LoopProtectionModule'} = 'Kernel::System::PostMaster::LoopProtection::DB';`

6.5.5. LoopProtectionLog

Description:	Path for the log file (it only applies if "FS" was selected for LoopProtectionModule and it is mandatory).
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'LoopProtectionLog'} = '<OTRS_CONFIG_Home>/var/log/LoopProtection';`

6.5.6. PostmasterAutoHTML2Text

Description:	Converts HTML mails into text messages.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterAutoHTML2Text'} = '1';`

6.5.7. PostmasterFollowUpSearchInReferences

Description:	Executes follow up checks on In-Reply-To or References headers for mails that don't have a ticket number in the subject.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterFollowUpSearchInReferences'} = '0';`

6.5.8. PostmasterFollowUpSearchInBody

Description:	Executes follow up mail body checks in mails that don't have a ticket number in the subject.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterFollowUpSearchInBody'} = '0';`

6.5.9. PostmasterFollowUpSearchInAttachment

Description:	Executes follow up mail attachments checks in mails that don't have a ticket number in the subject.
Group:	Ticket

SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterFollowUpSearchInAttachment'} = '0';`

6.5.10. PostmasterFollowUpSearchInRaw

Description:	Executes follow up plain/raw mail checks in mails that don't have a ticket number in the subject.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterFollowUpSearchInRaw'} = '0';`

6.5.11. PostmasterUserID

Description:	Specifies user id of the postmaster data base.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterUserID'} = '1';`

6.5.12. PostmasterDefaultQueue

Description:	Defines the postmaster default queue.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterDefaultQueue'} = 'Raw';`

6.5.13. PostmasterDefaultPriority

Description:	Defines the default priority of new tickets.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterDefaultPriority'} = '3 normal';`

6.5.14. PostmasterDefaultState

Description:	Defines the default state of new tickets.
Group:	Ticket

SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterDefaultState'} = 'new';`

6.5.15. PostmasterFollowUpState

Description:	Defines the state of a ticket if it gets a follow-up.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterFollowUpState'} = 'open';`

6.5.16. PostmasterFollowUpStateClosed

Description:	Defines the state of a ticket if it gets a follow-up and the ticket was already closed.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0
Config-Setting:	`$Self->{'PostmasterFollowUpStateClosed'} = 'open';`

6.5.17. PostmasterFollowUpOnUnlockAgentNotifyOnlyToOwner

Description:	Sends agent follow-up notification only to the owner, if a ticket is unlocked (the default is to send the notification to all agents).
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterFollowUpOnUnlockAgentNotifyOnlyToOwner'} = '0';`

6.5.18. PostmasterX-Header

Description:	Defines all the X-headers that should be scanned.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1

Config-Setting:	`$Self->{'PostmasterX-Header'} = [`
	`'From',`
	`'To',`
	`'Cc',`
	`'Reply-To',`
	`'ReplyTo',`

```
  'Subject',
  'Message-ID',
  'Message-Id',
  'Resent-To',
  'Resent-From',
  'Precedence',
  'Mailing-List',
  'List-Id',
  'List-Archive',
  'Errors-To',
  'References',
  'In-Reply-To',
  'Auto-Submitted',
  'X-Loop',
  'X-Spam-Flag',
  'X-Spam-Level',
  'X-Spam-Score',
  'X-Spam-Status',
  'X-No-Loop',
  'X-Priority',
  'Importance',
  'X-Mailer',
  'User-Agent',
  'Organization',
  'X-Original-To',
  'Delivered-To',
  'Envelope-To',
  'Return-Path',
  'X-OTRS-Owner',
  'X-OTRS-OwnerID',
  'X-OTRS-Responsible',
  'X-OTRS-ResponsibleID',
  'X-OTRS-Loop',
  'X-OTRS-Priority',
  'X-OTRS-Queue',
  'X-OTRS-Lock',
  'X-OTRS-Ignore',
  'X-OTRS-State',
  'X-OTRS-State-PendingTime',
  'X-OTRS-Type',
  'X-OTRS-Service',
  'X-OTRS-SLA',
  'X-OTRS-CustomerNo',
  'X-OTRS-CustomerUser',
  'X-OTRS-SenderType',
  'X-OTRS-ArticleType',
  'X-OTRS-FollowUp-Priority',
  'X-OTRS-FollowUp-Queue',
  'X-OTRS-FollowUp-Lock',
  'X-OTRS-FollowUp-State',
  'X-OTRS-FollowUp-State-PendingTime',
  'X-OTRS-FollowUp-Type',
  'X-OTRS-FollowUp-Service',
  'X-OTRS-FollowUp-SLA',
  'X-OTRS-FollowUp-SenderType',
  'X-OTRS-FollowUp-ArticleType'
];
```

6.5.19. PostMaster::PreFilterModule###1-Match

Description:	Module to filter and manipulate incoming messages. Block/ignore all spam email with From: noreply@ address.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0
Config-Setting:	`$Self->{'PostMaster::PreFilterModule'}->{'1-Match'} = {`

```
  'Match' => {
    'From' => 'noreply@'
  },
  'Module' => 'Kernel::System::PostMaster::Filter::Match',
  'Set' => {
    'X-OTRS-Ignore' => 'yes'
  },
  'StopAfterMatch' => '0'
};
```

6.5.20. PostMaster::PreFilterModule###2-Match

Description:	Module to filter and manipulate incoming messages. Get a 4 digit number to ticket free text, use regex in Match e. g. From => '(.+?)@.+?', and use () as [***] in Set =>.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0
Config-Setting:	`$Self->{'PostMaster::PreFilterModule'}->{'2-Match'} = { 'Match' => { 'Subject' => 'SomeNumber:(\\d\\d\\d\\d)' }, 'Module' => 'Kernel::System::PostMaster::Filter::Match', 'Set' => { 'X-OTRS-DynamicField-TicketFreeKey1' => 'SomeNumber', 'X-OTRS-DynamicField-TicketFreeText1' => '[***]' }, 'StopAfterMatch' => '0' };`

6.5.21. PostMaster::PreFilterModule###3-NewTicketReject

Description:	Blocks all the incoming emails that do not have a valid ticket number in subject with From: @example.com address.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0
Config-Setting:	`$Self->{'PostMaster::PreFilterModule'}->{'3-NewTicketReject'} = { 'Match' => { 'From' => '@example.com' }, 'Module' => 'Kernel::System::PostMaster::Filter::NewTicketReject', 'Set' => { 'X-OTRS-Ignore' => 'yes' }, 'StopAfterMatch' => '0' };`

6.5.22. PostMaster::PreFilterModule::NewTicketReject::Sender

Description:	Defines the sender for rejected emails.
Group:	Ticket
SubGroup:	Core::PostMaster

Valid:	0
Required:	0
Config-Setting:	`$Self->{'PostMaster::PreFilterModule::NewTicketReject::Sender'} = 'noreply@example.com';`

6.5.23. PostMaster::PreFilterModule::NewTicketReject::Subject

Description:	Defines the subject for rejected emails.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostMaster::PreFilterModule::NewTicketReject::Subject'} = 'Email Rejected';`

6.5.24. PostMaster::PreFilterModule::NewTicketReject::Body

Description:	Defines the body text for rejected emails.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1

Config-Setting:	`$Self->{'PostMaster::PreFilterModule::NewTicketReject::Body'} = '`
	`Dear Customer,`
	
	`Unfortunately we could not detect a valid ticket number`
	`in your subject, so this email can\'t be processed.`
	
	`Please create a new ticket via the customer panel.`
	
	`Thanks for your help!`
	
	`Your Helpdesk Team`
	`';`

6.5.25. PostMaster::PreFilterModule###4-CMD

Description:	CMD example setup. Ignores emails where external CMD returns some output on STDOUT (email will be piped into STDIN of some.bin).
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0

Config-Setting:	`$Self->{'PostMaster::PreFilterModule'}->{'4-CMD'} = {`
	`'CMD' => '/usr/bin/some.bin',`
	`'Module' => 'Kernel::System::PostMaster::Filter::CMD',`
	`'Set' => {`
	`'X-OTRS-Ignore' => 'yes'`
	`}`
	`};`

6.5.26. PostMaster::PreFilterModule###5-SpamAssassin

Description:	Spam Assassin example setup. Ignores emails that are marked with SpamAssassin.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0

Config-Setting:	`$Self->{'PostMaster::PreFilterModule'}->{'5-SpamAssassin'} = {`
	`  'CMD' => '/usr/bin/spamassassin | grep -i "X-Spam-Status: yes"',`
	`  'Module' => 'Kernel::System::PostMaster::Filter::CMD',`
	`  'Set' => {`
	`  'X-OTRS-Ignore' => 'yes'`
	`  }`
	`};`

6.5.27. PostMaster::PreFilterModule###6-SpamAssassin

Description:	Spam Assassin example setup. Moves marked mails to spam queue.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0

Config-Setting:	`$Self->{'PostMaster::PreFilterModule'}->{'6-SpamAssassin'} = {`
	`  'CMD' => '/usr/bin/spamassassin | grep -i "X-Spam-Status: yes"',`
	`  'Module' => 'Kernel::System::PostMaster::Filter::CMD',`
	`  'Set' => {`
	`  'X-OTRS-Queue' => 'spam'`
	`  }`
	`};`

6.5.28. PostMaster::PreFilterModule###000-MatchDBSource

Description:	Module to use database filter storage.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1

Config-Setting:	`$Self->{'PostMaster::PreFilterModule'}->{'000-MatchDBSource'} = {`
	`  'Module' => 'Kernel::System::PostMaster::Filter::MatchDBSource'`
	`};`

6.5.29. PostMaster::PostFilterModule###000-FollowUpArticleTypeCheck

Description:	Module to check if arrived emails should be marked as email-internal (because of original forwarded internal email). ArticleType and SenderType define the values for the arrived email/article.

Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'PostMaster::PostFilterModule'}->{'000-FollowUpArticleTypeCheck'} = {
	'ArticleType' => 'email-internal',
	'Module' =>
	'Kernel::System::PostMaster::Filter::FollowUpArticleTypeCheck',
	'SenderType' => 'customer'
	};</pre>

6.5.30. PostMaster::PreFilterModule###000-ExternalTicketNumberRecognition1

Description:	Recognize if a ticket is a follow up to an existing ticket using an external ticket number.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0

Config-Setting:	<pre>$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'
	};</pre>

6.5.31. PostMaster::PreFilterModule###000-ExternalTicketNumberRecognition2

Description:	Recognize if a ticket is a follow up to an existing ticket using an external ticket number.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0

Config-Setting:	<pre>$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',</pre>

```
  'TicketStateTypes' => 'new;open'
};
```

6.5.32. PostMaster::PreFilterModule###000-ExternalTicketNumberRecognition3

Description:	Recognize if a ticket is a follow up to an existing ticket using an external ticket number.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0

Config-Setting:	<pre>$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'
	};</pre>

6.5.33. PostMaster::PreFilterModule###000-ExternalTicketNumberRecognition4

Description:	Recognize if a ticket is a follow up to an existing ticket using an external ticket number.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0

Config-Setting:	<pre>$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'
	};</pre>

6.5.34. SendNoAutoResponseRegExp

Description:	If this regex matches, no message will be send by the autoresponder.
Group:	Ticket
SubGroup:	Core::PostMaster

Valid:	1
Required:	1
Config-Setting:	`$Self->{'SendNoAutoResponseRegExp'} = '(MAILER-DAEMON|postmaster|abuse)@.+?\\..+?';`

6.5.35. AutoResponseForWebTickets

Description:	If this option is set to 'Yes', tickets created via the web interface, via Customers or Agents, will receive an autoresponse if configured. If this option is set to 'No', no autoresponses will be sent.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'AutoResponseForWebTickets'} = '1';`

6.6. Core::Stats

6.6.1. Stats::DynamicObjectRegistration###Ticket

Description:	Module to generate ticket statistics.
Group:	Ticket
SubGroup:	Core::Stats
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Stats::DynamicObjectRegistration'}->{'Ticket'} = { 'Module' => 'Kernel::System::Stats::Dynamic::Ticket' };`

6.6.2. Stats::DynamicObjectRegistration###TicketList

Description:	Determines if the statistics module may generate ticket lists.
Group:	Ticket
SubGroup:	Core::Stats
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Stats::DynamicObjectRegistration'}->{'TicketList'} = { 'Module' => 'Kernel::System::Stats::Dynamic::TicketList' };`

6.6.3. Stats::DynamicObjectRegistration###TicketAccountedTime

Description:	Module to generate accounted time ticket statistics.
Group:	Ticket
SubGroup:	Core::Stats
Valid:	1

Required:	0
Config-Setting:	`$Self->{'Stats::DynamicObjectRegistration'}->{'TicketAccountedTime'} = { 'Module' => 'Kernel::System::Stats::Dynamic::TicketAccountedTime' };`

6.6.4. Stats::DynamicObjectRegistration###TicketSolutionResponseTi

Description:	Module to generate ticket solution and response time statistics.
Group:	Ticket
SubGroup:	Core::Stats
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Stats::DynamicObjectRegistration'}->{'TicketSolutionResponseTime'} = { 'Module' => 'Kernel::System::Stats::Dynamic::TicketSolutionResponseTime' };`

6.7. Core::Ticket

6.7.1. Ticket::Hook

Description:	The identifier for a ticket, e.g. Ticket#, Call#, MyTicket#. The default is Ticket#.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Hook'} = 'Ticket#';`

6.7.2. Ticket::HookDivider

Description:	The divider between TicketHook and ticket number. E.g ': '.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::HookDivider'} = '';`

6.7.3. Ticket::SubjectSize

Description:	Max size of the subjects in an email reply.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::SubjectSize'} = '100';`

6.7.4. Ticket::SubjectRe

Description:	The text at the beginning of the subject in an email reply, e.g. RE, AW, or AS.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::SubjectRe'} = 'Re';`

6.7.5. Ticket::SubjectFwd

Description:	The text at the beginning of the subject when an email is forwarded, e.g. FW, Fwd, or WG.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::SubjectFwd'} = 'Fwd';`

6.7.6. Ticket::SubjectFormat

Description:	The format of the subject. 'Left' means '[TicketHook#:12345] Some Subject', 'Right' means 'Some Subject [TicketHook#:12345]', 'None' means 'Some Subject' and no ticket number. In the last case you should enable PostmasterFollowupSearchInRaw or PostmasterFollowUpSearchInReferences to recognize followups based on email headers and/or body.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::SubjectFormat'} = 'Left';`

6.7.7. Ticket::CustomQueue

Description:	Name of custom queue. The custom queue is a queue selection of your preferred queues and can be selected in the preferences settings.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::CustomQueue'} = 'My Queues';`

6.7.8. Ticket::NewArticleIgnoreSystemSender

Description:	Ignore article with system sender type for new article feature (e. g. auto responses or email notifications).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::NewArticleIgnoreSystemSender'} = '0';`

6.7.9. Ticket::ChangeOwnerToEveryone

Description:	Changes the owner of tickets to everyone (useful for ASP). Normally only agent with rw permissions in the queue of the ticket will be shown.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::ChangeOwnerToEveryone'} = '0';`

6.7.10. Ticket::Responsible

Description:	Enables ticket responsible feature, to keep track of a specific ticket.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Responsible'} = '0';`

6.7.11. Ticket::ResponsibleAutoSet

Description:	Automatically sets the owner of a ticket as the responsible for it (if ticket responsible feature is enabled).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::ResponsibleAutoSet'} = '1';`

6.7.12. Ticket::Type

Description:	Allows defining new types for ticket (if ticket type feature is enabled).
Group:	Ticket

SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Type'} =  '0';`

6.7.13. Ticket::Service

Description:	Allows defining services and SLAs for tickets (e. g. email, desktop, network, ...), and escalation attributes for SLAs (if ticket service/SLA feature is enabled).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Service'} =  '0';`

6.7.14. Ticket::Service::Default::UnknownCustomer

Description:	Allows default services to be selected also for non existing customers.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Service::Default::UnknownCustomer'} =  '0';`

6.7.15. Ticket::ArchiveSystem

Description:	Activates the ticket archive system to have a faster system by moving some tickets out of the daily scope. To search for these tickets, the archive flag has to be enabled in the ticket search.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::ArchiveSystem'} =  '0';`

6.7.16. Ticket::ArchiveSystem::RemoveSeenFlags

Description:	Controls if the ticket and article seen flags are removed when a ticket is archived.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::ArchiveSystem::RemoveSeenFlags'} =  '1';`

6.7.17. Ticket::ArchiveSystem::RemoveTicketWatchers

Description:	Removes the ticket watcher information when a ticket is archived.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::ArchiveSystem::RemoveTicketWatchers'} = '1';`

6.7.18. Ticket::CustomerArchiveSystem

Description:	Activates the ticket archive system search in the customer interface.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::CustomerArchiveSystem'} = '0';`

6.7.19. Ticket::NumberGenerator

Description:	Selects the ticket number generator module. "AutoIncrement" increments the ticket number, the SystemID and the counter are used with SystemID.counter format (e.g. 1010138, 1010139). With "Date" the ticket numbers will be generated by the current date, the SystemID and the counter. The format looks like Year.Month.Day.SystemID.counter (e.g. 200206231010138, 200206231010139). With "DateChecksum" the counter will be appended as checksum to the string of date and SystemID. The checksum will be rotated on a daily basis. The format looks like Year.Month.Day.SystemID.Counter.CheckSum (e.g. 2002070110101520, 2002070110101535). "Random" generates randomized ticket numbers in the format "SystemID.Random" (e.g. 100057866352, 103745394596).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::NumberGenerator'} = 'Kernel::System::Ticket::Number::DateChecksum';`

6.7.20. Ticket::NumberGenerator::CheckSystemID

Description:	Checks the SystemID in ticket number detection for follow-ups (use "No" if SystemID has been changed after using the system).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::NumberGenerator::CheckSystemID'} = '1';`

6.7.21. Ticket::NumberGenerator::MinCounterSize

Description:	Sets the minimal ticket counter size (if "AutoIncrement" was selected as TicketNumberGenerator). Default is 5, this means the counter starts from 10000.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::NumberGenerator::MinCounterSize'} = '5';`

6.7.22. Ticket::NumberGenerator::Date::UseFormattedCounter

Description:	Enables the minimal ticket counter size (if "Date" was selected as TicketNumberGenerator).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::NumberGenerator::Date::UseFormattedCounter'} = '0';`

6.7.23. Ticket::CounterLog

Description:	Log file for the ticket counter.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::CounterLog'} = '<OTRS_CONFIG_Home>/var/log/TicketCounter.log';`

6.7.24. Ticket::IndexModule

Description:	IndexAccelerator: to choose your backend TicketViewAccelerator module. "RuntimeDB" generates each queue view on the fly from ticket table (no performance problems up to approx. 60.000 tickets in total and 6.000 open tickets in the system). "StaticDB" is the most powerful module, it uses an extra ticket-index table that works like a view (recommended if more than 80.000 and 6.000 open tickets are stored in the system). Use the script "bin/otrs.RebuildTicketIndex.pl" for initial index update.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1

Required:	1
Config-Setting:	`$Self->{'Ticket::IndexModule'} = 'Kernel::System::Ticket::IndexAccelerator::RuntimeDB';`

6.7.25. Ticket::StorageModule

Description:	Saves the attachments of articles. "DB" stores all data in the database (not recommended for storing big attachments). "FS" stores the data on the filesystem; this is faster but the webserver should run under the OTRS user. You can switch between the modules even on a system that is already in production without any loss of data.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::StorageModule'} = 'Kernel::System::Ticket::ArticleStorageDB';`

6.7.26. ArticleDir

Description:	Specifies the directory to store the data in, if "FS" was selected for TicketStorageModule.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'ArticleDir'} =  '<OTRS_CONFIG_Home>/var/article';`

6.7.27. Ticket::EventModulePost###100-ArchiveRestore

Description:	Restores a ticket from the archive (only if the event is a state change, from closed to any open available state).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'100-ArchiveRestore'} =  { 'Event' => 'TicketStateUpdate', 'Module' => 'Kernel::System::Ticket::Event::ArchiveRestore' };`

6.7.28. Ticket::EventModulePost###110-AcceleratorUpdate

Description:	Updates the ticket index accelerator.
Group:	Ticket
SubGroup:	Core::Ticket

Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'110-AcceleratorUpdate'} = { 'Event' => 'TicketStateUpdate|TicketQueueUpdate|TicketLockUpdate', 'Module' => 'Kernel::System::Ticket::Event::TicketAcceleratorUpdate' };`

6.7.29. Ticket::EventModulePost###120-ForceOwnerResetOnMove

Description:	Resets and unlocks the owner of a ticket if it was moved to another queue.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'120-ForceOwnerResetOnMove'} = { 'Event' => 'TicketQueueUpdate', 'Module' => 'Kernel::System::Ticket::Event::ForceOwnerReset' };`

6.7.30. Ticket::EventModulePost###130-ForceStateChangeOnLock

Description:	Forces to choose a different ticket state (from current) after lock action. Define the current state as key, and the next state after lock action as content.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'130-ForceStateChangeOnLock'} = { 'Event' => 'TicketLockUpdate', 'Module' => 'Kernel::System::Ticket::Event::ForceState', 'new' => 'open' };`

6.7.31. Ticket::EventModulePost###140-ResponsibleAutoSet

Description:	Automatically sets the responsible of a ticket (if it is not set yet) after the first owner update.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'140-ResponsibleAutoSet'} = { 'Event' => 'TicketOwnerUpdate', 'Module' => 'Kernel::System::Ticket::Event::ResponsibleAutoSet'`

	};

6.7.32. Ticket::EventModulePost###150-TicketPendingTimeReset

Description:	Sets the PendingTime of a ticket to 0 if the state is changed to a non-pending state.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'150-TicketPendingTimeReset'} = {`
	`'Event' => 'TicketStateUpdate',`
	`'Module' => 'Kernel::System::Ticket::Event::TicketPendingTimeReset'`
	`};`

6.7.33. Ticket::EventModulePost###500-NotificationEvent

Description:	Sends the notifications which are configured in the admin interface under "Notfication (Event)".
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'500-NotificationEvent'} = {`
	`'Event' => '',`
	`'Module' => 'Kernel::System::Ticket::Event::NotificationEvent',`
	`'Transaction' => '1'`
	`};`

6.7.34. Ticket::EventModulePost###900-EscalationIndex

Description:	Updates the ticket escalation index after a ticket attribute got updated.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'900-EscalationIndex'} = {`
	`'Event' => 'TicketSLAUpdate|TicketQueueUpdate|TicketStateUpdate|TicketCreate|ArticleCreate',`
	`'Module' => 'Kernel::System::Ticket::Event::TicketEscalationIndex'`
	`};`

6.7.35. Ticket::EventModulePost###900-EscalationStopEvents

Description:	Ticket event module that triggers the escalation stop events.

Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'900-EscalationStopEvents'} = { 'Event' => 'TicketSLAUpdate|TicketQueueUpdate|TicketStateUpdate|ArticleCreate', 'Module' => 'Kernel::System::Ticket::Event::TriggerEscalationStopEvents' };`

6.7.36. Ticket::EventModulePost###910-ForceUnlockOnMove

Description:	Forces to unlock tickets after being moved to another queue.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'910-ForceUnlockOnMove'} = { 'Event' => 'TicketQueueUpdate', 'Module' => 'Kernel::System::Ticket::Event::ForceUnlock' };`

6.7.37. Ticket::EventModulePost###920-TicketArticleNewMessageUpdate

Description:	Update Ticket "Seen" flag if every article got seen or a new Article got created.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'920-TicketArticleNewMessageUpdate'} = { 'Event' => 'ArticleCreate|ArticleFlagSet', 'Module' => 'Kernel::System::Ticket::Event::TicketNewMessageUpdate' };`

6.7.38. DynamicFieldFromCustomerUser::Mapping

Description:	Define a mapping between variables of the customer user data (keys) and dynamic fields of a ticket (values). The purpose is to store customer user data in ticket dynamic fields. The dynamic fields must be present in the system and should be enabled for AgentTicketFreeText, so that they can be set/updated manually by the agent. They mustn't be enabled for AgentTicketPhone, AgentTicketEmail and AgentTicketCustomer. If they were, they would have precedence over the automatically set values. To use this mapping, you have to also activate the next setting below.
Group:	Ticket

SubGroup:	Core::Ticket
Valid:	0
Required:	0

Config-Setting:	`$Self->{'DynamicFieldFromCustomerUser::Mapping'} = {`
	`  'UserFirstname' => 'CustomerFirstname'`
	`};`

6.7.39. Ticket::EventModulePost###930-DynamicFieldFromCustomerUser

Description:	This event module stores attributes from CustomerUser as DynamicFields tickets. Please see the setting above for how to configure the mapping.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'930-DynamicFieldFromCustomerUser'} = {`
	`  'Event' => '(TicketCreate|TicketCustomerUpdate)',`
	`  'Module' =>`
	`  'Kernel::System::Ticket::Event::DynamicFieldFromCustomerUser'`
	`};`

6.7.40. Ticket::CustomModule###001-CustomModule

Description:	Overloads (redefines) existing functions in Kernel::System::Ticket. Used to easily add customizations.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::CustomModule'}->{'001-CustomModule'} =` `'Kernel::System::Ticket::Custom';`

6.7.41. Ticket::ViewableSenderTypes

Description:	Defines the default viewable sender types of a ticket (default: customer).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::ViewableSenderTypes'} = [`
	`  '\'customer\''`
	`];`

6.7.42. Ticket::ViewableLocks

Description:	Defines the viewable locks of a ticket. Default: unlock, tmp_lock.

Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::ViewableLocks'} = [ '\'unlock\'', '\'tmp_lock\'' ];`

6.7.43. Ticket::ViewableStateType

Description:	Defines the valid state types for a ticket.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::ViewableStateType'} = [ 'new', 'open', 'pending reminder', 'pending auto' ];`

6.7.44. Ticket::UnlockStateType

Description:	Defines the valid states for unlocked tickets. To unlock tickets the script "bin/otrs.UnlockTickets.pl" can be used.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::UnlockStateType'} = [ 'new', 'open' ];`

6.7.45. Ticket::PendingNotificationOnlyToOwner

Description:	Sends reminder notifications of unlocked ticket after reaching the reminder date (only sent to ticket owner).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::PendingNotificationOnlyToOwner'} = '0';`

6.7.46. Ticket::PendingNotificationNotToResponsible

Description:	Disables sending reminder notifications to the responsible agent of a ticket (Ticket::Responsible needs to be activated).
Group:	Ticket

SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::PendingNotificationNotToResponsible'} = '0';`

6.7.47. Ticket::PendingReminderStateType

Description:	Defines the state type of the reminder for pending tickets.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::PendingReminderStateType'} = [ 'pending reminder' ];`

6.7.48. Ticket::PendingAutoStateType

Description:	Determines the possible states for pending tickets that changed state after reaching time limit.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::PendingAutoStateType'} = [ 'pending auto' ];`

6.7.49. Ticket::StateAfterPending

Description:	Defines which states should be set automatically (Content), after the pending time of state (Key) has been reached.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::StateAfterPending'} = { 'pending auto close+' => 'closed successful', 'pending auto close-' => 'closed unsuccessful' };`

6.7.50. System::Permission

Description:	Standard available permissions for agents within the application. If more permissions are needed, they can be entered here. Permissions must be defined to be effective. Some other good permissions have also been provided built-in: note, close, pending, customer, freetext, move, compose, responsible, forward, and bounce. Make sure that "rw" is always the last registered permission.

Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'System::Permission'} = [ 'ro', 'move_into', 'create', 'note', 'owner', 'priority', 'rw' ];`

6.7.51. Ticket::Permission###1-OwnerCheck

Description:	Module to check the owner of a ticket.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Permission'}->{'1-OwnerCheck'} = { 'Granted' => '1', 'Module' => 'Kernel::System::Ticket::Permission::OwnerCheck', 'Required' => '0' };`

6.7.52. Ticket::Permission###2-ResponsibleCheck

Description:	Module to check the agent responsible of a ticket.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Permission'}->{'2-ResponsibleCheck'} = { 'Granted' => '1', 'Module' => 'Kernel::System::Ticket::Permission::ResponsibleCheck', 'Required' => '0' };`

6.7.53. Ticket::Permission###3-GroupCheck

Description:	Module to check if a user is in a special group. Access is granted, if the user is in the specified group and has ro and rw permissions.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Permission'}->{'3-GroupCheck'} = { 'Granted' => '1', 'Module' => 'Kernel::System::Ticket::Permission::GroupCheck', 'Required' => '0' };`

6.7.54. Ticket::Permission###4-WatcherCheck

Description:	Module to check the watcher agents of a ticket.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Permission'}->{'4-WatcherCheck'} = { 'Granted' => '1', 'Module' => 'Kernel::System::Ticket::Permission::WatcherCheck', 'Required' => '0' };`

6.7.55. CustomerTicket::Permission###1-GroupCheck

Description:	Module to check the group permissions for the access to customer tickets.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerTicket::Permission'}->{'1-GroupCheck'} = { 'Granted' => '0', 'Module' => 'Kernel::System::Ticket::CustomerPermission::GroupCheck', 'Required' => '1' };`

6.7.56. CustomerTicket::Permission###2-CustomerUserIDCheck

Description:	Grants access, if the customer ID of the ticket matches the customer user's ID and the customer user has group permissions on the queue the ticket is in.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerTicket::Permission'}->{'2-CustomerUserIDCheck'} = { 'Granted' => '1', 'Module' => 'Kernel::System::Ticket::CustomerPermission::CustomerUserIDCheck', 'Required' => '0' };`

6.7.57. CustomerTicket::Permission###3-CustomerIDCheck

Description:	Module to check customer permissions.
Group:	Ticket
SubGroup:	Core::Ticket

Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerTicket::Permission'}->{'3-CustomerIDCheck'} = { 'Granted' => '1', 'Module' => 'Kernel::System::Ticket::CustomerPermission::CustomerIDCheck', 'Required' => '0' };`

6.7.58. Ticket::DefineEmailFrom

Description:	Defines how the From field from the emails (sent from answers and email tickets) should look like.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::DefineEmailFrom'} = 'SystemAddressName';`

6.7.59. Ticket::DefineEmailFromSeparator

Description:	Defines the separator between the agents real name and the given queue email address.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::DefineEmailFromSeparator'} = 'via';`

6.7.60. CustomerNotifyJustToRealCustomer

Description:	Sends customer notifications just to the mapped customer. Normally, if no customer is mapped, the latest customer sender gets the notification.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerNotifyJustToRealCustomer'} = '0';`

6.7.61. AgentSelfNotifyOnAction

Description:	Specifies if an agent should receive email notification of his own actions.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1

Config-Setting:	`$Self->{'AgentSelfNotifyOnAction'} = '0';`

6.7.62. Ticket::EventModulePost###900-GenericAgent

Description:	Event module registration. For more performance you can define a trigger event (e. g. Event => TicketCreate).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'900-GenericAgent'} = { 'Event' => '', 'Module' => 'Kernel::System::Ticket::Event::GenericAgent', 'Transaction' => '1' };`

6.7.63. StandardTemplate::Types

Description:	Defines the list of types for templates.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'StandardTemplate::Types'} = { 'Answer' => 'Answer', 'Create' => 'Create', 'Forward' => 'Forward', 'PhoneCall' => 'Phone call' };`

6.8. Core::TicketACL

6.8.1. Ticket::Acl::Module###1-Ticket::Acl::Module

Description:	ACL module that allows closing parent tickets only if all its children are already closed ("State" shows which states are not available for the parent ticket until all child tickets are closed).
Group:	Ticket
SubGroup:	Core::TicketACL
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Acl::Module'}->{'1-Ticket::Acl::Module'} = { 'Module' => 'Kernel::System::Ticket::Acl::CloseParentAfterClosedChilds', 'State' => [ 'closed successful', 'closed unsuccessful' ] };`

6.8.2. TicketACL::Default::Action

Description:	Default ACL values for ticket actions.

Group:	Ticket
SubGroup:	Core::TicketACL
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TicketACL::Default::Action'} = {};`

6.8.3. ACLKeysLevel1Match

Description:	Defines which items are available in first level of the ACL structure.
Group:	Ticket
SubGroup:	Core::TicketACL
Valid:	1
Required:	0
Config-Setting:	`$Self->{'ACLKeysLevel1Match'} = { 'Properties' => 'Properties', 'PropertiesDatabase' => 'PropertiesDatabase' };`

6.8.4. ACLKeysLevel1Change

Description:	Defines which items are available in first level of the ACL structure.
Group:	Ticket
SubGroup:	Core::TicketACL
Valid:	1
Required:	0
Config-Setting:	`$Self->{'ACLKeysLevel1Change'} = { 'Possible' => 'Possible', 'PossibleNot' => 'PossibleNot' };`

6.8.5. ACLKeysLevel2::Possible

Description:	Defines which items are available in second level of the ACL structure.
Group:	Ticket
SubGroup:	Core::TicketACL
Valid:	1
Required:	0
Config-Setting:	`$Self->{'ACLKeysLevel2::Possible'} = { 'Action' => 'Action', 'ActivityDialog' => 'ActivityDialog', 'Process' => 'Process', 'Ticket' => 'Ticket' };`

6.8.6. ACLKeysLevel2::PossibleNot

Description:	Defines which items are available in second level of the ACL structure.

Group:	Ticket
SubGroup:	Core::TicketACL
Valid:	1
Required:	0

Config-Setting:	`$Self->{'ACLKeysLevel2::PossibleNot'} = {`
	`'ActivityDialog' => 'ActivityDialog',`
	`'Process' => 'Process',`
	`'Ticket' => 'Ticket'`
	`};`

6.8.7. ACLKeysLevel2::Properties

Description:	Defines which items are available in second level of the ACL structure.
Group:	Ticket
SubGroup:	Core::TicketACL
Valid:	1
Required:	0

Config-Setting:	`$Self->{'ACLKeysLevel2::Properties'} = {`
	`'CustomerUser' => 'CustomerUser',`
	`'DynamicField' => 'DynamicField',`
	`'Frontend' => 'Frontend',`
	`'Owner' => 'Owner',`
	`'Priority' => 'Priority',`
	`'Process' => 'Process',`
	`'Queue' => 'Queue',`
	`'Responsible' => 'Responsible',`
	`'SLA' => 'SLA',`
	`'Service' => 'Service',`
	`'State' => 'State',`
	`'Ticket' => 'Ticket',`
	`'Type' => 'Type',`
	`'User' => 'User'`
	`};`

6.8.8. ACLKeysLevel2::PropertiesDatabase

Description:	Defines which items are available in second level of the ACL structure.
Group:	Ticket
SubGroup:	Core::TicketACL
Valid:	1
Required:	0

Config-Setting:	`$Self->{'ACLKeysLevel2::PropertiesDatabase'} = {`
	`'CustomerUser' => 'CustomerUser',`
	`'DynamicField' => 'DynamicField',`
	`'Owner' => 'Owner',`
	`'Priority' => 'Priority',`
	`'Process' => 'Process',`
	`'Queue' => 'Queue',`
	`'Responsible' => 'Responsible',`
	`'SLA' => 'SLA',`
	`'Service' => 'Service',`
	`'State' => 'State',`
	`'Ticket' => 'Ticket',`
	`'Type' => 'Type',`
	`'User' => 'User'`
	`};`

6.8.9. ACLKeysLevel3::Actions###100-Default

Description:	Defines which items are available for 'Action' in third level of the ACL structure.
Group:	Ticket
SubGroup:	Core::TicketACL
Valid:	1
Required:	0
Config-Setting:	`$Self->{'ACLKeysLevel3::Actions'}->{'100-Default'} = ['AgentTicketBounce', 'AgentTicketClose', 'AgentTicketCompose', 'AgentTicketCustomer', 'AgentTicketForward', 'AgentTicketFreeText', 'AgentTicketHistory', 'AgentTicketLink', 'AgentTicketLock', 'AgentTicketMerge', 'AgentTicketMove', 'AgentTicketNote', 'AgentTicketOwner', 'AgentTicketPending', 'AgentTicketPhone', 'AgentTicketPhoneInbound', 'AgentTicketPhoneOutbound', 'AgentTicketPrint', 'AgentTicketPriority', 'AgentTicketResponsible', 'AgentTicketWatcher', 'AgentTicketZoom', 'AgentLinkObject', 'CustomerTicketMessage', 'CustomerTicketPrint', 'CustomerTicketProcess', 'CustomerTicketZoom'];`

6.8.10. ACL::CacheTTL

Description:	Cache time in seconds for the DB ACL backend.
Group:	Ticket
SubGroup:	Core::TicketACL
Valid:	1
Required:	1
Config-Setting:	`$Self->{'ACL::CacheTTL'} = '3600';`

6.9. Core::TicketBulkAction

6.9.1. Ticket::Frontend::BulkFeature

Description:	Enables ticket bulk action feature for the agent frontend to work on more than one ticket at a time.
Group:	Ticket
SubGroup:	Core::TicketBulkAction
Valid:	1

Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::BulkFeature'} = '1';`

6.9.2. Ticket::Frontend::BulkFeatureGroup

Description:	Enables ticket bulk action feature only for the listed groups.
Group:	Ticket
SubGroup:	Core::TicketBulkAction
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::BulkFeatureGroup'} = [ 'admin', 'users' ];`

6.10. Core::TicketDynamicFieldDefault

6.10.1. Ticket::EventModulePost###TicketDynamicFieldDefault

Description:	Event module registration. For more performance you can define a trigger event (e. g. Event => TicketCreate). This is only possible if all Ticket dynamic fields need the same event.
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'TicketDynamicFieldDefault'} = { 'Module' => 'Kernel::System::Ticket::Event::TicketDynamicFieldDefault', 'Transaction' => '1' };`

6.10.2. Ticket::TicketDynamicFieldDefault###Element1

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element1'} = { 'Event' => 'TicketCreate', 'Name' => 'Field1', 'Value' => 'Default' };`

6.10.3. Ticket::TicketDynamicFieldDefault###Element2

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element2'} = {`
	`'Event' => '',`
	`'Name' => '',`
	`'Value' => ''`
	`};`

6.10.4. Ticket::TicketDynamicFieldDefault###Element3

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element3'} = {`
	`'Event' => '',`
	`'Name' => '',`
	`'Value' => ''`
	`};`

6.10.5. Ticket::TicketDynamicFieldDefault###Element4

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element4'} = {`
	`'Event' => '',`
	`'Name' => '',`
	`'Value' => ''`

	};

6.10.6. Ticket::TicketDynamicFieldDefault###Element5

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element5'} = {`
	`'Event' => '',`
	`'Name' => '',`
	`'Value' => ''`
	`};`

6.10.7. Ticket::TicketDynamicFieldDefault###Element6

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element6'} = {`
	`'Event' => '',`
	`'Name' => '',`
	`'Value' => ''`
	`};`

6.10.8. Ticket::TicketDynamicFieldDefault###Element7

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element7'} = {` `'Event' => '',`

	'Name' => '', 'Value' => '' };

6.10.9. Ticket::TicketDynamicFieldDefault###Element8

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0

Config-Setting:	$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element8'} = {
	'Event' => '',
	'Name' => '',
	'Value' => ''
	};

6.10.10. Ticket::TicketDynamicFieldDefault###Element9

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0

Config-Setting:	$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element9'} = {
	'Event' => '',
	'Name' => '',
	'Value' => ''
	};

6.10.11. Ticket::TicketDynamicFieldDefault###Element10

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element10'} = { 'Event' => '', 'Name' => '', 'Value' => '' };`

6.10.12. Ticket::TicketDynamicFieldDefault###Element11

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element11'} = { 'Event' => '', 'Name' => '', 'Value' => '' };`

6.10.13. Ticket::TicketDynamicFieldDefault###Element12

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element12'} = { 'Event' => '', 'Name' => '', 'Value' => '' };`

6.10.14. Ticket::TicketDynamicFieldDefault###Element13

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0

Required:	0

Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element13'} = {`
	`'Event' => '',`
	`'Name' => '',`
	`'Value' => ''`
	`};`

6.10.15. Ticket::TicketDynamicFieldDefault###Element14

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element14'} = {`
	`'Event' => '',`
	`'Name' => '',`
	`'Value' => ''`
	`};`

6.10.16. Ticket::TicketDynamicFieldDefault###Element15

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element15'} = {`
	`'Event' => '',`
	`'Name' => '',`
	`'Value' => ''`
	`};`

6.10.17. Ticket::TicketDynamicFieldDefault###Element16

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault

Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element16'} = {`
	`'Event' => '',`
	`'Name' => '',`
	`'Value' => ''`
	`};`

6.11. Core::TicketWatcher

6.11.1. Ticket::Watcher

Description:	Enables or disables the ticket watcher feature, to keep track of tickets without being the owner nor the responsible.
Group:	Ticket
SubGroup:	Core::TicketWatcher
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Watcher'} = '0';`

6.11.2. Ticket::WatcherGroup

Description:	Enables ticket watcher feature only for the listed groups.
Group:	Ticket
SubGroup:	Core::TicketWatcher
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::WatcherGroup'} = [`
	`'admin',`
	`'users'`
	`];`

6.12. Frontend::Admin

6.12.1. Events###Ticket

Description:	List of all ticket events to be displayed in the GUI.
Group:	Ticket
SubGroup:	Frontend::Admin
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Events'}->{'Ticket'} = [`
	`'TicketCreate',`
	`'TicketDelete',`
	`'TicketTitleUpdate',`
	`'TicketUnlockTimeoutUpdate',`
	`'TicketQueueUpdate',`
	`'TicketTypeUpdate',`
	`'TicketServiceUpdate',`
	`'TicketSLAUpdate',`
	`'TicketCustomerUpdate',`
	`'TicketPendingTimeUpdate',`

```
  'TicketLockUpdate',
  'TicketArchiveFlagUpdate',
  'TicketStateUpdate',
  'TicketOwnerUpdate',
  'TicketResponsibleUpdate',
  'TicketPriorityUpdate',
  'HistoryAdd',
  'HistoryDelete',
  'TicketAccountTime',
  'TicketMerge',
  'TicketSubscribe',
  'TicketUnsubscribe',
  'TicketFlagSet',
  'TicketFlagDelete',
  'TicketSlaveLinkAdd',
  'TicketSlaveLinkDelete',
  'TicketMasterLinkDelete',
  'EscalationResponseTimeNotifyBefore',
  'EscalationUpdateTimeNotifyBefore',
  'EscalationSolutionTimeNotifyBefore',
  'EscalationResponseTimeStart',
  'EscalationUpdateTimeStart',
  'EscalationSolutionTimeStart',
  'EscalationResponseTimeStop',
  'EscalationUpdateTimeStop',
  'EscalationSolutionTimeStop'
];
```

6.12.2. Events###Article

Description:	List of all article events to be displayed in the GUI.
Group:	Ticket
SubGroup:	Frontend::Admin
Valid:	1
Required:	1
Config-Setting:	

```
 $Self->{'Events'}->{'Article'} =  [
  'ArticleCreate',
  'ArticleUpdate',
  'ArticleSend',
  'ArticleBounce',
  'ArticleAgentNotification',
  'ArticleCustomerNotification',
  'ArticleAutoResponse',
  'ArticleFlagSet',
  'ArticleFlagDelete',
  'ArticleAgentNotification',
  'ArticleCustomerNotification'
];
```

6.13. Frontend::Admin::ModuleRegistration

6.13.1. Frontend::Module###AdminACL

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

```
 $Self->{'Frontend::Module'}->{'AdminACL'} =  {
  'Description' => 'Admin',
  'Group' => [
```

```
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.ACL.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.ACL.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'Ticket',
    'Description' => 'Configure and manage ACLs.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Access Control Lists (ACL)',
    'Prio' => '750'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Access Control Lists (ACL)'
};
```

6.13.2. Frontend::Module###AdminQueue

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

```
 $Self->{'Frontend::Module'}->{'AdminQueue'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Queue',
    'Description' => 'Create and manage queues.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Queues',
    'Prio' => '100'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Queues'
};
```

6.13.3. Frontend::Module###AdminTemplate

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

```
 $Self->{'Frontend::Module'}->{'AdminTemplate'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Queue',
    'Description' => 'Create and manage templates.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Templates',
    'Prio' => '200'
```

```
  },
  'NavBarName' => 'Admin',
  'Title' => 'Templates'
};
```

6.13.4. Frontend::Module###AdminQueueTemplates

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

```
 $Self->{'Frontend::Module'}->{'AdminQueueTemplates'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Queue',
    'Description' => 'Link templates to queues.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Templates <-> Queues',
    'Prio' => '300'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Templates <-> Queues'
};
```

6.13.5. Frontend::Module###AdminAutoResponse

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

```
 $Self->{'Frontend::Module'}->{'AdminAutoResponse'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Queue',
    'Description' => 'Create and manage responses that are
 automatically sent.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Auto Responses',
    'Prio' => '400'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Auto Responses'
};
```

6.13.6. Frontend::Module###AdminQueueAutoResponse

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration

Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AdminQueueAutoResponse'} = {
	'Description' => 'Admin',
	'Group' => [
	'admin'
	],
	'NavBarModule' => {
	'Block' => 'Queue',
	'Description' => 'Link queues to auto responses.',
	'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
	'Name' => 'Auto Responses <-> Queues',
	'Prio' => '500'
	},
	'NavBarName' => 'Admin',
	'Title' => 'Auto Responses <-> Queues'
	};</pre>

6.13.7. Frontend::Module###AdminAttachment

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

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AdminAttachment'} = {
	'Description' => 'Admin',
	'Group' => [
	'admin'
	],
	'NavBarModule' => {
	'Block' => 'Queue',
	'Description' => 'Create and manage attachments.',
	'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
	'Name' => 'Attachments',
	'Prio' => '600'
	},
	'NavBarName' => 'Admin',
	'Title' => 'Attachments'
	};</pre>

6.13.8. Frontend::Module###AdminTemplateAttachment

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

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AdminTemplateAttachment'} = {
	'Description' => 'Admin',
	'Group' => [
	'admin'
	],
	'NavBarModule' => {
	'Block' => 'Queue',
	'Description' => 'Link attachments to templates.',
	'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
	'Name' => 'Attachments <-> Templates',
	'Prio' => '700'</pre>

```
  },
  'NavBarName' => 'Admin',
  'Title' => 'Attachments <-> Templates'
};
```

6.13.9. Frontend::Module###AdminSalutation

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

```
 $Self->{'Frontend::Module'}->{'AdminSalutation'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Queue',
    'Description' => 'Create and manage salutations.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Salutations',
    'Prio' => '800'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Salutations'
};
```

6.13.10. Frontend::Module###AdminSignature

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

```
 $Self->{'Frontend::Module'}->{'AdminSignature'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Queue',
    'Description' => 'Create and manage signatures.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Signatures',
    'Prio' => '900'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Signatures'
};
```

6.13.11. Frontend::Module###AdminSystemAddress

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

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AdminSystemAddress'} = {
	'Description' => 'Admin',
	'Group' => [
	'admin'
	],
	'NavBarModule' => {
	'Block' => 'Email',
	'Description' => 'Set sender email addresses for this system.',
	'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
	'Name' => 'Email Addresses',
	'Prio' => '300'
	},
	'NavBarName' => 'Admin',
	'Title' => 'Email Addresses'
	};</pre>

6.13.12. Frontend::Module###AdminNotification

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

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AdminNotification'} = {
	'Description' => 'Admin',
	'Group' => [
	'admin'
	],
	'NavBarModule' => {
	'Block' => 'Ticket',
	'Description' => 'Manage notifications that are sent to
	agents.',
	'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
	'Name' => 'Agent Notifications',
	'Prio' => '400'
	},
	'NavBarName' => 'Admin',
	'Title' => 'Agent Notifications'
	};</pre>

6.13.13. Frontend::Module###AdminNotificationEvent

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

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AdminNotificationEvent'} = {
	'Description' => 'Admin',
	'Group' => [
	'admin'
	],
	'NavBarModule' => {
	'Block' => 'Ticket',
	'Description' => 'Create and manage event based notifications.',
	'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
	'Name' => 'Notifications (Event)',
	'Prio' => '400'
	},
	'NavBarName' => 'Admin',
	'Title' => 'Notifications (Event)'
	};</pre>

6.13.14. Frontend::Module###AdminService

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

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AdminService'} = {
	  'Description' => 'Admin',
	  'Group' => [
	    'admin'
	  ],
	  'NavBarModule' => {
	    'Block' => 'Ticket',
	    'Description' => 'Create and manage services.',
	    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
	    'Name' => 'Services',
	    'Prio' => '900'
	  },
	  'NavBarName' => 'Admin',
	  'Title' => 'Services'
	};</pre>

6.13.15. Frontend::Module###AdminSLA

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

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AdminSLA'} = {
	  'Description' => 'Admin',
	  'Group' => [
	    'admin'
	  ],
	  'NavBarModule' => {
	    'Block' => 'Ticket',
	    'Description' => 'Create and manage Service Level Agreements
	(SLAs).',
	    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
	    'Name' => 'Service Level Agreements',
	    'Prio' => '1000'
	  },
	  'NavBarName' => 'Admin',
	  'Title' => 'Service Level Agreements'
	};</pre>

6.13.16. Frontend::Module###AdminType

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

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AdminType'} = {
	  'Description' => 'Admin',
	  'Group' => [
	    'admin'</pre>

```
  ],
  'NavBarModule' => {
    'Block' => 'Ticket',
    'Description' => 'Create and manage ticket types.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Types',
    'Prio' => '700'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Types'
};
```

6.13.17. Frontend::Module###AdminState

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

```
 $Self->{'Frontend::Module'}->{'AdminState'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Ticket',
    'Description' => 'Create and manage ticket states.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'States',
    'Prio' => '800'
  },
  'NavBarName' => 'Admin',
  'Title' => 'States'
};
```

6.13.18. Frontend::Module###AdminPriority

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

```
 $Self->{'Frontend::Module'}->{'AdminPriority'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Ticket',
    'Description' => 'Create and manage ticket priorities.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Priorities',
    'Prio' => '850'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Priorities'
};
```

6.13.19. Frontend::Module###AdminGenericAgent

Description:	Frontend module registration for the agent interface.

Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AdminGenericAgent'} = {
	  'Description' => 'Admin',
	  'Group' => [
	    'admin'
	  ],
	  'Loader' => {
	    'JavaScript' => [
	      'Core.Agent.Admin.GenericAgent.js'
	    ]
	  },
	  'NavBarModule' => {
	    'Block' => 'System',
	    'Description' => 'Manage tasks triggered by event or time based execution.',
	    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
	    'Name' => 'GenericAgent',
	    'Prio' => '300'
	  },
	  'NavBarName' => 'Admin',
	  'Title' => 'GenericAgent'
	};</pre>

6.14. Frontend::Agent

6.14.1. Ticket::Frontend::PendingDiffTime

Description:	Time in seconds that gets added to the actual time if setting a pending-state (default: 86400 = 1 day).
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<pre>$Self->{'Ticket::Frontend::PendingDiffTime'} = '86400';</pre>

6.14.2. Ticket::Frontend::MaxQueueLevel

Description:	Define the max depth of queues.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<pre>$Self->{'Ticket::Frontend::MaxQueueLevel'} = '5';</pre>

6.14.3. Ticket::Frontend::ListType

Description:	Shows existing parent/child queue lists in the system in the form of a tree or a list.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1

Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ListType'} = 'tree';`

6.14.4. Ticket::Frontend::TextAreaEmail

Description:	Permitted width for compose email windows.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::TextAreaEmail'} = '82';`

6.14.5. Ticket::Frontend::TextAreaNote

Description:	Permitted width for compose note windows.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::TextAreaNote'} = '78';`

6.14.6. Ticket::Frontend::InformAgentMaxSize

Description:	Max size (in rows) of the informed agents box in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::InformAgentMaxSize'} = '3';`

6.14.7. Ticket::Frontend::InvolvedAgentMaxSize

Description:	Max size (in rows) of the involved agents box in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::InvolvedAgentMaxSize'} = '3';`

6.14.8. Ticket::Frontend::CustomerInfoCompose

Description:	Shows the customer user information (phone and email) in the compose screen.
Group:	Ticket

SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerInfoCompose'} = '1';`

6.14.9. Ticket::Frontend::CustomerInfoComposeMaxSize

Description:	Max size (in characters) of the customer information table (phone and email) in the compose screen.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerInfoComposeMaxSize'} = '22';`

6.14.10. Ticket::Frontend::CustomerInfoZoom

Description:	Shows the customer user's info in the ticket zoom view.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerInfoZoom'} = '1';`

6.14.11. Ticket::Frontend::CustomerInfoZoomMaxSize

Description:	Maximum size (in characters) of the customer information table in the ticket zoom view.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerInfoZoomMaxSize'} = '22';`

6.14.12. Ticket::Frontend::DynamicFieldsZoomMaxSizeSidebar

Description:	Maximum length (in characters) of the dynamic field in the sidebar of the ticket zoom view.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::DynamicFieldsZoomMaxSizeSidebar'} = '18';`

6.14.13. Ticket::Frontend::DynamicFieldsZoomMaxSizeArticle

Description:	Maximum length (in characters) of the dynamic field in the article of the ticket zoom view.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::DynamicFieldsZoomMaxSizeArticle'} = '160';`

6.14.14. Ticket::Frontend::AccountTime

Description:	Activates time accounting.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AccountTime'} = '1';`

6.14.15. Ticket::Frontend::TimeUnits

Description:	Sets the prefered time units (e.g. work units, hours, minutes).
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::TimeUnits'} = ' (work units)';`

6.14.16. Ticket::Frontend::NeedAccountedTime

Description:	Defines if time accounting is mandatory in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::NeedAccountedTime'} = '0';`

6.14.17. Ticket::Frontend::BulkAccountedTime

Description:	Defines if time accounting must be set to all tickets in bulk action.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::BulkAccountedTime'} = '1';`

6.14.18. Ticket::Frontend::NeedSpellCheck

Description:	Defines if composed messages have to be spell checked in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::NeedSpellCheck'} = '0';`

6.14.19. Ticket::Frontend::NewOwnerSelection

Description:	Shows an owner selection in phone and email tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::NewOwnerSelection'} = '1';`

6.14.20. Ticket::Frontend::NewResponsibleSelection

Description:	Show a responsible selection in phone and email tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::NewResponsibleSelection'} = '1';`

6.14.21. Ticket::Frontend::NewQueueSelectionType

Description:	Defines the receipent target of the phone ticket and the sender of the email ticket ("Queue" shows all queues, "SystemAddress" displays all system addresses) in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::NewQueueSelectionType'} = 'Queue';`

6.14.22. Ticket::Frontend::NewQueueSelectionString

Description:	Determines the strings that will be shown as receipent (To:) of the phone ticket and as sender (From:) of the email ticket in the agent

	interface. For Queue as NewQueueSelectionType "<Queue>" shows the names of the queues and for SystemAddress "<Realname> <<Email>>" shows the name and email of the receipent.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::NewQueueSelectionString'} = '<Queue>';`

6.14.23. Ticket::Frontend::NewQueueOwnSelection

Description:	Determines which options will be valid of the recepient (phone ticket) and the sender (email ticket) in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::NewQueueOwnSelection'} = { '1' => 'First Queue!', '2' => 'Second Queue!' };`

6.14.24. Ticket::Frontend::ShowCustomerTickets

Description:	Shows customer history tickets in AgentTicketPhone, AgentTicketEmail and AgentTicketCustomer.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ShowCustomerTickets'} = '1';`

6.14.25. NewTicketInNewWindow::Enabled

Description:	If enabled, TicketPhone and TicketEmail will be open in new windows.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'NewTicketInNewWindow::Enabled'} = '0';`

6.14.26. CustomerDBLink

Description:	Defines an external link to the database of the customer (e.g. 'http://yourhost/customer.php?CID=$Data{"CustomerID"}' or '').
Group:	Ticket

SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerDBLink'} = '$Env{"CGIHandle"}?Action=AgentCustomerInformationCenter;CustomerID=$LQData{"CustomerID"}';`

6.14.27. CustomerDBLinkTarget

Description:	Defines the target attribute in the link to external customer database. E.g. 'target="cdb"'.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerDBLinkTarget'} = '';`

6.14.28. CustomerDBLinkClass

Description:	Defines the target attribute in the link to external customer database. E.g. 'AsPopup PopupType_TicketAction'.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerDBLinkClass'} = '';`

6.14.29. Frontend::CommonObject###QueueObject

Description:	Path of the file that stores all the settings for the QueueObject object for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::CommonObject'}->{'QueueObject'} = 'Kernel::System::Queue';`

6.14.30. Frontend::CommonObject###TicketObject

Description:	Path of the file that stores all the settings for the TicketObject for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::CommonObject'}->{'TicketObject'} = 'Kernel::System::Ticket';`

6.14.31. Frontend::CommonParam###Action

Description:	Defines the default used Frontend-Module if no Action parameter given in the url on the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::CommonParam'}->{'Action'} = 'AgentDashboard';`

6.14.32. Frontend::CommonParam###QueueID

Description:	Default queue ID used by the system in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::CommonParam'}->{'QueueID'} = '0';`

6.14.33. Frontend::CommonParam###TicketID

Description:	Default ticket ID used by the system in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::CommonParam'}->{'TicketID'} = '';`

6.14.34. DefaultOverviewColumns

Description:	General ticket data shown in the ticket overviews (fall-back). Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default. Note that TicketNumber can not be disabled, because it is necessary.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	

```
$Self->{'DefaultOverviewColumns'} =  {
 'Age' => '2',
 'Changed' => '1',
 'CustomerID' => '2',
 'CustomerName' => '1',
 'CustomerUserID' => '1',
 'EscalationResponseTime' => '1',
 'EscalationSolutionTime' => '1',
 'EscalationTime' => '1',
 'EscalationUpdateTime' => '1',
 'Lock' => '2',
 'Owner' => '2',
 'PendingTime' => '1',
 'Priority' => '1',
```

```
  'Queue' => '2',
  'Responsible' => '1',
  'SLA' => '1',
  'Service' => '1',
  'State' => '2',
  'TicketNumber' => '2',
  'Title' => '2',
  'Type' => '1'
};
```

6.15. Frontend::Agent::Dashboard

6.15.1. DashboardBackend###0100-TicketPendingReminder

Description:	Parameters for the dashboard backend of the ticket pending reminder overview of the agent interface . "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed for DefaultColumns. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'DashboardBackend'}->{'0100-TicketPendingReminder'} =  {
 'Attributes' => 'TicketPendingTimeOlderMinutes=1;StateType=pending
reminder;SortBy=PendingTime;OrderBy=Down;',
 'Block' => 'ContentLarge',
 'CacheTTLLocal' => '0.5',
 'Default' => '1',
 'DefaultColumns' => {
   'Age' => '2',
   'Changed' => '1',
   'CustomerID' => '1',
   'CustomerName' => '1',
   'CustomerUserID' => '1',
   'EscalationResponseTime' => '1',
   'EscalationSolutionTime' => '1',
   'EscalationTime' => '1',
   'EscalationUpdateTime' => '1',
   'Lock' => '1',
   'Owner' => '1',
   'PendingTime' => '1',
   'Priority' => '1',
   'Queue' => '1',
   'Responsible' => '1',
   'SLA' => '1',
   'Service' => '1',
   'State' => '1',
   'TicketNumber' => '2',
   'Title' => '2',
   'Type' => '1'
 },
 'Description' => 'All tickets with a reminder set where the
reminder date has been reached',
 'Filter' => 'Locked',
 'Group' => '',
 'Limit' => '10',
```

```
  'Module' => 'Kernel::Output::HTML::DashboardTicketGeneric',
  'Permission' => 'rw',
  'Time' => 'UntilTime',
  'Title' => 'Reminder Tickets'
};
```

6.15.2. DashboardBackend###0110-TicketEscalation

Description:	Parameters for the dashboard backend of the ticket escalation overview of the agent interface . "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed for DefaultColumns. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'DashboardBackend'}->{'0110-TicketEscalation'} =  {
  'Attributes' =>
 'TicketEscalationTimeOlderMinutes=1;SortBy=EscalationTime;OrderBy=Down;',
  'Block' => 'ContentLarge',
  'CacheTTLLocal' => '0.5',
  'Default' => '1',
  'DefaultColumns' => {
    'Age' => '2',
    'Changed' => '1',
    'CustomerID' => '1',
    'CustomerName' => '1',
    'CustomerUserID' => '1',
    'EscalationResponseTime' => '1',
    'EscalationSolutionTime' => '1',
    'EscalationTime' => '1',
    'EscalationUpdateTime' => '1',
    'Lock' => '1',
    'Owner' => '1',
    'PendingTime' => '1',
    'Priority' => '1',
    'Queue' => '1',
    'Responsible' => '1',
    'SLA' => '1',
    'Service' => '1',
    'State' => '1',
    'TicketNumber' => '2',
    'Title' => '2',
    'Type' => '1'
  },
  'Description' => 'All escalated tickets',
  'Filter' => 'All',
  'Group' => '',
  'Limit' => '10',
  'Module' => 'Kernel::Output::HTML::DashboardTicketGeneric',
  'Permission' => 'rw',
  'Time' => 'EscalationTime',
  'Title' => 'Escalated Tickets'
};
```

6.15.3. DashboardBackend###0120-TicketNew

Description:	Parameters for the dashboard backend of the new tickets overview of the agent interface. "Limit" is the number of entries shown by

	default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed for DefaultColumns. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'DashboardBackend'}->{'0120-TicketNew'} = {
	  'Attributes' => 'StateType=new;',
	  'Block' => 'ContentLarge',
	  'CacheTTLLocal' => '0.5',
	  'Default' => '1',
	  'DefaultColumns' => {
	    'Age' => '2',
	    'Changed' => '1',
	    'CustomerID' => '1',
	    'CustomerName' => '1',
	    'CustomerUserID' => '1',
	    'EscalationResponseTime' => '1',
	    'EscalationSolutionTime' => '1',
	    'EscalationTime' => '1',
	    'EscalationUpdateTime' => '1',
	    'Lock' => '1',
	    'Owner' => '1',
	    'PendingTime' => '1',
	    'Priority' => '1',
	    'Queue' => '1',
	    'Responsible' => '1',
	    'SLA' => '1',
	    'Service' => '1',
	    'State' => '1',
	    'TicketNumber' => '2',
	    'Title' => '2',
	    'Type' => '1'
	  },
	  'Description' => 'All new tickets, these tickets have not been
	 worked on yet',
	  'Filter' => 'All',
	  'Group' => '',
	  'Limit' => '10',
	  'Module' => 'Kernel::Output::HTML::DashboardTicketGeneric',
	  'Permission' => 'rw',
	  'Time' => 'Age',
	  'Title' => 'New Tickets'
	};</pre>

6.15.4. DashboardBackend###0130-TicketOpen

Description:	Parameters for the dashboard backend of the ticket pending reminder overview of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed for DefaultColumns. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default.
Group:	Ticket

SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'DashboardBackend'}->{'0130-TicketOpen'} = {
	  'Attributes' => 'StateType=open;',
	  'Block' => 'ContentLarge',
	  'CacheTTLLocal' => '0.5',
	  'Default' => '1',
	  'DefaultColumns' => {
	    'Age' => '2',
	    'Changed' => '1',
	    'CustomerID' => '1',
	    'CustomerName' => '1',
	    'CustomerUserID' => '1',
	    'EscalationResponseTime' => '1',
	    'EscalationSolutionTime' => '1',
	    'EscalationTime' => '1',
	    'EscalationUpdateTime' => '1',
	    'Lock' => '1',
	    'Owner' => '1',
	    'PendingTime' => '1',
	    'Priority' => '1',
	    'Queue' => '1',
	    'Responsible' => '1',
	    'SLA' => '1',
	    'Service' => '1',
	    'State' => '1',
	    'TicketNumber' => '2',
	    'Title' => '2',
	    'Type' => '1'
	  },
	  'Description' => 'All open tickets, these tickets have already
	 been worked on, but need a response',
	  'Filter' => 'All',
	  'Group' => '',
	  'Limit' => '10',
	  'Module' => 'Kernel::Output::HTML::DashboardTicketGeneric',
	  'Permission' => 'rw',
	  'Time' => 'Age',
	  'Title' => 'Open Tickets / Need to be answered'
	};</pre>

6.15.5. DashboardBackend###0250-TicketStats

Description:	Parameters for the dashboard backend of the ticket stats of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'DashboardBackend'}->{'0250-TicketStats'} = {
	  'Block' => 'ContentSmall',
	  'CacheTTL' => '30',
	  'Changed' => '1',
	  'Closed' => '1',
	  'Default' => '1',
	  'Group' => '',
	  'Module' => 'Kernel::Output::HTML::DashboardTicketStatsGeneric',
	  'Permission' => 'rw',
	  'Title' => '7 Day Stats'</pre>

```
};
```

6.15.6. DashboardBackend###0260-TicketCalendar

Description:	Parameters for the dashboard backend of the ticket calendar of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'DashboardBackend'}->{'0260-TicketCalendar'} = {
	'Block' => 'ContentSmall',
	'CacheTTL' => '2',
	'Default' => '1',
	'Group' => '',
	'Limit' => '6',
	'Module' => 'Kernel::Output::HTML::DashboardCalendar',
	'OwnerOnly' => '',
	'Permission' => 'rw',
	'Title' => 'Upcoming Events'
	};</pre>

6.15.7. DashboardBackend###0270-TicketQueueOverview

Description:	Parameters for the dashboard backend of the queue overview widget of the agent interface. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "QueuePermissionGroup" is not mandatory, queues are only listed if they belong to this permission group if you enable it. "States" is a list of states, the key is the sort order of the state in the widget. "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'DashboardBackend'}->{'0270-TicketQueueOverview'} = {
	'Block' => 'ContentLarge',
	'CacheTTLLocal' => '0.5',
	'Default' => '1',
	'Description' => 'Provides a matrix overview of the tickets per state per queue.',
	'Group' => '',
	'Module' => 'Kernel::Output::HTML::DashboardTicketQueueOverview',
	'Permission' => 'rw',
	'QueuePermissionGroup' => 'users',
	'Sort' => 'SortBy=Age;OrderBy=Up',
	'States' => {
	'1' => 'new',
	'4' => 'open',
	'6' => 'pending reminder'
	},</pre>

```
  'Title' => 'Ticket Queue Overview'
};
```

6.15.8. DashboardBackend###0280-DashboardEventsTicketCalendar

Description:	Parameters for the dashboard backend of the ticket stats of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0

Config-Setting:	`$Self->{'DashboardBackend'}->{'0280-DashboardEventsTicketCalendar'} = {`
	`'Block' => 'ContentLarge',`
	`'CacheTTL' => '0',`
	`'Default' => '0',`
	`'Group' => '',`
	`'Module' => 'Kernel::Output::HTML::DashboardEventsTicketCalendar',`
	`'Title' => 'Events Ticket Calendar'`
	`};`

6.15.9. AgentCustomerInformationCenter::Backend###0100-CIC-TicketPendingReminder

Description:	Parameters for the dashboard backend of the ticket pending reminder overview of the agent interface . "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed for DefaultColumns. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0

Config-Setting:	`$Self->{'AgentCustomerInformationCenter::Backend'}->{'0100-CIC-TicketPendingReminder'} = {`
	`'Attributes' => 'TicketPendingTimeOlderMinutes=1;StateType=pendingreminder;SortBy=PendingTime;OrderBy=Down;',`
	`'Block' => 'ContentLarge',`
	`'CacheTTLLocal' => '0.5',`
	`'Default' => '1',`
	`'DefaultColumns' => {`
	`'Age' => '2',`
	`'Changed' => '1',`
	`'CustomerID' => '1',`
	`'CustomerName' => '1',`

```
    'CustomerUserID' => '1',
    'EscalationResponseTime' => '1',
    'EscalationSolutionTime' => '1',
    'EscalationTime' => '1',
    'EscalationUpdateTime' => '1',
    'Lock' => '1',
    'Owner' => '1',
    'PendingTime' => '1',
    'Priority' => '1',
    'Queue' => '1',
    'Responsible' => '1',
    'SLA' => '1',
    'Service' => '1',
    'State' => '1',
    'TicketNumber' => '2',
    'Title' => '2',
    'Type' => '1'
  },
  'Description' => 'All tickets with a reminder set where the
 reminder date has been reached',
  'Filter' => 'Locked',
  'Group' => '',
  'Limit' => '10',
  'Module' => 'Kernel::Output::HTML::DashboardTicketGeneric',
  'Permission' => 'ro',
  'Time' => 'UntilTime',
  'Title' => 'Reminder Tickets'
};
```

6.15.10. AgentCustomerInformationCenter::Backend###0110-CIC-TicketEscalation

Description:	Parameters for the dashboard backend of the ticket escalation overview of the agent interface . "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed for DefaultColumns. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'AgentCustomerInformationCenter::Backend'}->{'0110-CIC-
TicketEscalation'} =  {
  'Attributes' =>
 'TicketEscalationTimeOlderMinutes=1;SortBy=EscalationTime;OrderBy=Down;',
  'Block' => 'ContentLarge',
  'CacheTTLLocal' => '0.5',
  'Default' => '1',
  'DefaultColumns' => {
    'Age' => '2',
    'Changed' => '1',
    'CustomerID' => '1',
    'CustomerName' => '1',
    'CustomerUserID' => '1',
    'EscalationResponseTime' => '1',
    'EscalationSolutionTime' => '1',
    'EscalationTime' => '1',
    'EscalationUpdateTime' => '1',
    'Lock' => '1',
```

```
    'Owner' => '1',
    'PendingTime' => '1',
    'Priority' => '1',
    'Queue' => '1',
    'Responsible' => '1',
    'SLA' => '1',
    'Service' => '1',
    'State' => '1',
    'TicketNumber' => '2',
    'Title' => '2',
    'Type' => '1'
  },
  'Description' => 'All escalated tickets',
  'Filter' => 'All',
  'Group' => '',
  'Limit' => '10',
  'Module' => 'Kernel::Output::HTML::DashboardTicketGeneric',
  'Permission' => 'ro',
  'Time' => 'EscalationTime',
  'Title' => 'Escalated Tickets'
};
```

6.15.11. AgentCustomerInformationCenter::Backend###0120-CIC-TicketNew

Description:	Parameters for the dashboard backend of the new tickets overview of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed for DefaultColumns. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'AgentCustomerInformationCenter::Backend'}->{'0120-CIC-
TicketNew'} =  {
  'Attributes' => 'StateType=new;',
  'Block' => 'ContentLarge',
  'CacheTTLLocal' => '0.5',
  'Default' => '1',
  'DefaultColumns' => {
    'Age' => '2',
    'Changed' => '1',
    'CustomerID' => '1',
    'CustomerName' => '1',
    'CustomerUserID' => '1',
    'EscalationResponseTime' => '1',
    'EscalationSolutionTime' => '1',
    'EscalationTime' => '1',
    'EscalationUpdateTime' => '1',
    'Lock' => '1',
    'Owner' => '1',
    'PendingTime' => '1',
    'Priority' => '1',
    'Queue' => '1',
    'Responsible' => '1',
    'SLA' => '1',
    'Service' => '1',
    'State' => '1',
```

```
    'TicketNumber' => '2',
    'Title' => '2',
    'Type' => '1'
  },
  'Description' => 'All new tickets, these tickets have not been
 worked on yet',
  'Filter' => 'All',
  'Group' => '',
  'Limit' => '10',
  'Module' => 'Kernel::Output::HTML::DashboardTicketGeneric',
  'Permission' => 'ro',
  'Time' => 'Age',
  'Title' => 'New Tickets'
};
```

6.15.12. AgentCustomerInformationCenter::Backend###0130-CIC-TicketOpen

Description:	Parameters for the dashboard backend of the ticket pending reminder overview of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed for DefaultColumns. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'AgentCustomerInformationCenter::Backend'}->{'0130-CIC-
TicketOpen'} =  {
  'Attributes' => 'StateType=open;',
  'Block' => 'ContentLarge',
  'CacheTTLLocal' => '0.5',
  'Default' => '1',
  'DefaultColumns' => {
    'Age' => '2',
    'Changed' => '1',
    'CustomerID' => '1',
    'CustomerName' => '1',
    'CustomerUserID' => '1',
    'EscalationResponseTime' => '1',
    'EscalationSolutionTime' => '1',
    'EscalationTime' => '1',
    'EscalationUpdateTime' => '1',
    'Lock' => '1',
    'Owner' => '1',
    'PendingTime' => '1',
    'Priority' => '1',
    'Queue' => '1',
    'Responsible' => '1',
    'SLA' => '1',
    'Service' => '1',
    'State' => '1',
    'TicketNumber' => '2',
    'Title' => '2',
    'Type' => '1'
  },
  'Description' => 'All open tickets, these tickets have already
 been worked on, but need a response',
```

```
  'Filter' => 'All',
  'Group' => '',
  'Limit' => '10',
  'Module' => 'Kernel::Output::HTML::DashboardTicketGeneric',
  'Permission' => 'ro',
  'Time' => 'Age',
  'Title' => 'Open Tickets / Need to be answered'
};
```

6.15.13. AgentCustomerInformationCenter::Backend###0500-CIC-CustomerIDStatus

Description:	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.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'AgentCustomerInformationCenter::Backend'}->{'0500-CIC-
CustomerIDStatus'} =  {
  'Attributes' => '',
  'Block' => 'ContentSmall',
  'CacheTTLLocal' => '0.5',
  'Default' => '1',
  'Description' => 'Company Status',
  'Group' => '',
  'Module' => 'Kernel::Output::HTML::DashboardCustomerIDStatus',
  'Permission' => 'ro',
  'Title' => 'Company Status'
};
```

6.16. Frontend::Agent::Dashboard::EventsTicketCalendar

6.16.1. DashboardEventsTicketCalendar###CalendarWidth

Description:	Defines the calendar width in percent. Default is 95%.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard::EventsTicketCalendar
Valid:	1
Required:	1
Config-Setting:	

```
 $Self->{'DashboardEventsTicketCalendar'}->{'CalendarWidth'} =
 '95';
```

6.16.2. DashboardEventsTicketCalendar###Queues

Description:	Defines queues that's tickets are used for displaying as calendar events.

Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard::EventsTicketCalendar
Valid:	1
Required:	1
Config-Setting:	`$Self->{'DashboardEventsTicketCalendar'}->{'Queues'} = [ 'Raw' ];`

6.16.3. DashboardEventsTicketCalendar::DynamicFieldStartTime

Description:	Define dynamic field name for start time. This field has to be manually added to the system as Ticket: "Date / Time" and must be activated in ticket creation screens and/or in any other ticket action screens.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard::EventsTicketCalendar
Valid:	1
Required:	0
Config-Setting:	`$Self->{'DashboardEventsTicketCalendar::DynamicFieldStartTime'} = 'TicketCalendarStartTime';`

6.16.4. DashboardEventsTicketCalendar::DynamicFieldEndTime

Description:	Define dynamic field name for end time. This field has to be manually added to the system as Ticket: "Date / Time" and must be activated in ticket creation screens and/or in any other ticket action screens.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard::EventsTicketCalendar
Valid:	1
Required:	0
Config-Setting:	`$Self->{'DashboardEventsTicketCalendar::DynamicFieldEndTime'} = 'TicketCalendarEndTime';`

6.16.5. DashboardEventsTicketCalendar::DynamicFieldsForEvents

Description:	Defines the dynamic fields that are used for displaying on calendar events.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard::EventsTicketCalendar
Valid:	1
Required:	1
Config-Setting:	`$Self->{'DashboardEventsTicketCalendar::DynamicFieldsForEvents'} = [ 'TicketCalendarStartTime', 'TicketCalendarEndTime' ];`

6.16.6. DashboardEventsTicketCalendar::TicketFieldsForEvents

Description:	Defines the ticket fields that are going to be displayed calendar events. The "Key" defines the field or ticket attribute and the "Content" defines the display name.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard::EventsTicketCalendar
Valid:	1
Required:	1

Config-Setting:	`$Self->{'DashboardEventsTicketCalendar::TicketFieldsForEvents'} = {`
	`'CustomerID' => 'Customer ID',`
	`'CustomerUserID' => 'Customer user',`
	`'Priority' => 'Priority',`
	`'Queue' => 'Queue',`
	`'SLA' => 'SLA',`
	`'Service' => 'Service',`
	`'State' => 'State',`
	`'Title' => 'Title',`
	`'Type' => 'Type'`
	`};`

6.17. Frontend::Agent::Dashboard::TicketFilters

6.17.1. OnlyValuesOnTicket

Description:	Defines if the list for filters should be retrieve just from current tickets in system. Just for clarification, Customers list will always came from system's tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard::TicketFilters
Valid:	1
Required:	1
Config-Setting:	`$Self->{'OnlyValuesOnTicket'} = '1';`

6.18. Frontend::Agent::ModuleMetaHead

6.18.1. Frontend::HeaderMetaModule###2-TicketSearch

Description:	Module to generate html OpenSearch profile for short ticket search in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleMetaHead
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::HeaderMetaModule'}->{'2-TicketSearch'} = {`
	`'Action' => 'AgentTicketSearch',`
	`'Module' => 'Kernel::Output::HTML::HeaderMetaTicketSearch'`

```
};
```

6.19. Frontend::Agent::ModuleNotify

6.19.1. Frontend::NotifyModule###5-Ticket::TicketEscalation

Description:	Module to show notifications and escalations (ShownMax: max. shown escalations, EscalationInMinutes: Show ticket which will escalation in, CacheTime: Cache of calculated escalations in seconds).
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleNotify
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Frontend::NotifyModule'}->{'5-Ticket::TicketEscalation'} = { 'CacheTime' => '40', 'EscalationInMinutes' => '120', 'Module' => 'Kernel::Output::HTML::NotificationAgentTicketEscalation', 'ShownMax' => '25' };`

6.20. Frontend::Agent::ModuleRegistration

6.20.1. Frontend::Module###AgentTicketQueue

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketQueue'} = { 'Description' => 'Overview of all open Tickets', 'Loader' => { 'CSS' => ['Core.AgentTicketQueue.css', 'Core.AllocationList.css'], 'JavaScript' => ['Core.UI.AllocationList.js', 'Core.Agent.TableFilters.js'] }, 'NavBar' => [{ 'AccessKey' => 'o', 'Block' => '', 'Description' => 'Overview of all open Tickets', 'Link' => 'Action=AgentTicketQueue', 'LinkOption' => '', 'Name' => 'Queue view', 'NavBar' => 'Ticket', 'Prio' => '100', 'Type' => '' }, { 'AccessKey' => 't',`

```
      'Block' => 'ItemArea',
      'Description' => '',
      'Link' => 'Action=AgentTicketQueue',
      'LinkOption' => '',
      'Name' => 'Tickets',
      'NavBar' => 'Ticket',
      'Prio' => '200',
      'Type' => 'Menu'
    }
  ],
  'NavBarName' => 'Ticket',
  'Title' => 'QueueView'
};
```

6.20.2. Frontend::Module###AgentTicketPhone

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

```
 $Self->{'Frontend::Module'}->{'AgentTicketPhone'} =  {
  'Description' => 'Create new phone ticket',
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.CustomerSearch.js',
      'Core.Agent.TicketAction.js'
    ]
  },
  'NavBar' => [
    {
      'AccessKey' => 'n',
      'Block' => '',
      'Description' => 'Create new phone ticket (inbound)',
      'Link' => 'Action=AgentTicketPhone',
      'LinkOption' => '',
      'Name' => 'New phone ticket',
      'NavBar' => 'Ticket',
      'Prio' => '200',
      'Type' => ''
    }
  ],
  'NavBarName' => 'Ticket',
  'Title' => 'New phone ticket'
};
```

6.20.3. Frontend::Module###AgentTicketPhoneOutbound

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

```
 $Self->{'Frontend::Module'}->{'AgentTicketPhoneOutbound'} =  {
  'Description' => 'Phone Call',
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.TicketAction.js'
    ]
  },
  'NavBarName' => 'Ticket',
```

	`  'Title' => 'Phone-Ticket'` `};`

6.20.4. Frontend::Module###AgentTicketPhoneInbound

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketPhoneInbound'} =  {`
	`  'Description' => 'Incoming Phone Call',`
	`  'Loader' => {`
	`    'JavaScript' => [`
	`      'Core.Agent.TicketAction.js'`
	`    ]`
	`  },`
	`  'NavBarName' => 'Ticket',`
	`  'Title' => 'Phone-Ticket'`
	`};`

6.20.5. Frontend::Module###AgentTicketEmail

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketEmail'} =  {`
	`  'Description' => 'Create new email ticket',`
	`  'Loader' => {`
	`    'JavaScript' => [`
	`      'Core.Agent.CustomerSearch.js',`
	`      'Core.Agent.TicketAction.js'`
	`    ]`
	`  },`
	`  'NavBar' => [`
	`    {`
	`      'AccessKey' => 'm',`
	`      'Block' => '',`
	`      'Description' => 'Create new email ticket and send this out (outbound)',`
	`      'Link' => 'Action=AgentTicketEmail',`
	`      'LinkOption' => '',`
	`      'Name' => 'New email ticket',`
	`      'NavBar' => 'Ticket',`
	`      'Prio' => '210',`
	`      'Type' => ''`
	`    }`
	`  ],`
	`  'NavBarName' => 'Ticket',`
	`  'Title' => 'New email ticket'`
	`};`

6.20.6. Frontend::Module###AgentTicketSearch

Description:	Frontend module registration for the agent interface.
Group:	Ticket

SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AgentTicketSearch'} = {
	'Description' => 'Search Ticket',
	'Loader' => {
	'JavaScript' => [
	'Core.UI.AllocationList.js',
	'Core.Agent.TableFilters.js'
	]
	},
	'NavBar' => [
	{
	'AccessKey' => 's',
	'Block' => '',
	'Description' => 'Search Tickets',
	'Link' => 'Action=AgentTicketSearch',
	'LinkOption' => 'onclick="window.setTimeout(function()
	{Core.Agent.Search.OpenSearchDialog(\'AgentTicketSearch\');}, 0);
	return false;"',
	'Name' => 'Search',
	'NavBar' => 'Ticket',
	'Prio' => '300',
	'Type' => ''
	}
	],
	'NavBarName' => 'Ticket',
	'Title' => 'Search'
	};</pre>

6.20.7. Frontend::Module###AgentTicketLockedView

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	<pre>$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'
	};</pre>

6.20.8. Frontend::Module###AgentTicketResponsibleView

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1

Required:	0

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AgentTicketResponsibleView'} = {
	'Description' => 'Responsible Tickets',
	'Loader' => {
	'CSS' => [
	'Core.AllocationList.css'
	],
	'JavaScript' => [
	'Core.UI.AllocationList.js',
	'Core.Agent.TableFilters.js'
	]
	},
	'NavBarName' => 'Ticket',
	'Title' => 'Responsible Tickets'
	};</pre>

6.20.9. Frontend::Module###AgentTicketWatchView

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	<pre>$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'
	};</pre>

6.20.10. Frontend::Module###AgentCustomerSearch

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AgentCustomerSearch'} = {
	'Description' => 'AgentCustomerSearch',
	'NavBarName' => 'Ticket',
	'Title' => 'AgentCustomerSearch'
	};</pre>

6.20.11. Frontend::Module###AgentUserSearch

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1

Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentUserSearch'} = {`
	`'Description' => 'AgentUserSearch',`
	`'NavBarName' => 'Ticket',`
	`'Title' => 'AgentUserSearch'`
	`};`

6.20.12. Frontend::Module###AgentTicketStatusView

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketStatusView'} = {`
	`'Description' => 'Overview of all open tickets',`
	`'Loader' => {`
	`'CSS' => [`
	`'Core.AllocationList.css'`
	`],`
	`'JavaScript' => [`
	`'Core.UI.AllocationList.js',`
	`'Core.Agent.TableFilters.js'`
	`]`
	`},`
	`'NavBar' => [`
	`{`
	`'AccessKey' => 'v',`
	`'Block' => '',`
	`'Description' => 'Overview of all open Tickets.',`
	`'Link' => 'Action=AgentTicketStatusView',`
	`'LinkOption' => '',`
	`'Name' => 'Status view',`
	`'NavBar' => 'Ticket',`
	`'Prio' => '110',`
	`'Type' => ''`
	`}`
	`],`
	`'NavBarName' => 'Ticket',`
	`'Title' => 'Status view'`
	`};`

6.20.13. Frontend::Module###AgentTicketEscalationView

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketEscalationView'} = {`
	`'Description' => 'Overview of all escalated tickets',`
	`'Loader' => {`
	`'CSS' => [`
	`'Core.AllocationList.css'`
	`],`
	`'JavaScript' => [`
	`'Core.UI.AllocationList.js',`
	`'Core.Agent.TableFilters.js'`
	`]`
	`},`

```
  'NavBar' => [
    {
      'AccessKey' => 'e',
      'Block' => '',
      'Description' => 'Overview Escalated Tickets',
      'Link' => 'Action=AgentTicketEscalationView',
      'LinkOption' => '',
      'Name' => 'Escalation view',
      'NavBar' => 'Ticket',
      'Prio' => '120',
      'Type' => ''
    }
  ],
  'NavBarName' => 'Ticket',
  'Title' => 'Escalation view'
};
```

6.20.14. Frontend::Module###AgentZoom

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentZoom'} = {`
	`'Description' => 'compat module for AgentZoom to AgentTicketZoom',`
	`'NavBarName' => 'Ticket',`
	`'Title' => ''`
	`};`

6.20.15. Frontend::Module###AgentTicketZoom

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketZoom'} = {`
	`'Description' => 'Ticket Zoom',`
	`'Loader' => {`
	`'CSS' => [`
	`'Core.Agent.TicketProcess.css'`
	`],`
	`'JavaScript' => [`
	`'thirdparty/jquery-tablesorter-2.0.5/jquery.tablesorter.js',`
	`'Core.UI.Table.Sort.js',`
	`'Core.Agent.TicketZoom.js'`
	`]`
	`},`
	`'NavBarName' => 'Ticket',`
	`'Title' => 'Zoom'`
	`};`

6.20.16. Frontend::Module###AgentTicketAttachment

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1

Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketAttachment'} = {`
	`'Description' => 'To download attachments',`
	`'NavBarName' => 'Ticket',`
	`'Title' => ''`
	`};`

6.20.17. Frontend::Module###AgentTicketPlain

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketPlain'} = {`
	`'Description' => 'Ticket plain view of an email',`
	`'NavBarName' => 'Ticket',`
	`'Title' => 'Plain'`
	`};`

6.20.18. Frontend::Module###AgentTicketNote

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketNote'} = {`
	`'Description' => 'Ticket Note',`
	`'Loader' => {`
	`'JavaScript' => [`
	`'Core.Agent.TicketAction.js'`
	`]`
	`},`
	`'NavBarName' => 'Ticket',`
	`'Title' => 'Note'`
	`};`

6.20.19. Frontend::Module###AgentTicketMerge

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketMerge'} = {`
	`'Description' => 'Ticket Merge',`
	`'NavBarName' => 'Ticket',`
	`'Title' => 'Merge'`
	`};`

6.20.20. Frontend::Module###AgentTicketPending

Description:	Frontend module registration for the agent interface.

Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketPending'} = {`
	`'Description' => 'Ticket Pending',`
	`'Loader' => {`
	`'JavaScript' => [`
	`'Core.Agent.TicketAction.js'`
	`]`
	`},`
	`'NavBarName' => 'Ticket',`
	`'Title' => 'Pending'`
	`};`

6.20.21. Frontend::Module###AgentTicketWatcher

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketWatcher'} = {`
	`'Description' => 'A TicketWatcher Module',`
	`'NavBarName' => 'Ticket-Watcher',`
	`'Title' => 'Ticket-Watcher'`
	`};`

6.20.22. Frontend::Module###AgentTicketPriority

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketPriority'} = {`
	`'Description' => 'Ticket Priority',`
	`'Loader' => {`
	`'JavaScript' => [`
	`'Core.Agent.TicketAction.js'`
	`]`
	`},`
	`'NavBarName' => 'Ticket',`
	`'Title' => 'Priority'`
	`};`

6.20.23. Frontend::Module###AgentTicketLock

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketLock'} = {` `'Description' => 'Ticket Lock',`

```
  'NavBarName' => 'Ticket',
  'Title' => 'Lock'
};
```

6.20.24. Frontend::Module###AgentTicketMove

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketMove'} = {`
	`  'Description' => 'Ticket Move',`
	`  'Loader' => {`
	`  'JavaScript' => [`
	`  'Core.Agent.TicketAction.js'`
	`  ]`
	`  },`
	`  'NavBarName' => 'Ticket',`
	`  'Title' => 'Move'`
	`};`

6.20.25. Frontend::Module###AgentTicketHistory

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketHistory'} = {`
	`  'Description' => 'Ticket History',`
	`  'NavBarName' => 'Ticket',`
	`  'Title' => 'History'`
	`};`

6.20.26. Frontend::Module###AgentTicketOwner

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketOwner'} = {`
	`  'Description' => 'Ticket Owner',`
	`  'Loader' => {`
	`  'JavaScript' => [`
	`  'Core.Agent.TicketAction.js'`
	`  ]`
	`  },`
	`  'NavBarName' => 'Ticket',`
	`  'Title' => 'Owner'`
	`};`

6.20.27. Frontend::Module###AgentTicketResponsible

Description:	Frontend module registration for the agent interface.

Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AgentTicketResponsible'} = {
	'Description' => 'Ticket Responsible',
	'Loader' => {
	'JavaScript' => [
	'Core.Agent.TicketAction.js'
	]
	},
	'NavBarName' => 'Ticket',
	'Title' => 'Responsible'
	};</pre>

6.20.28. Frontend::Module###AgentTicketCompose

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AgentTicketCompose'} = {
	'Description' => 'Ticket Compose email Answer',
	'Loader' => {
	'JavaScript' => [
	'Core.Agent.CustomerSearch.js',
	'Core.Agent.TicketAction.js'
	]
	},
	'NavBarName' => 'Ticket',
	'Title' => 'Compose'
	};</pre>

6.20.29. Frontend::Module###AgentTicketBounce

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AgentTicketBounce'} = {
	'Description' => 'Ticket Compose Bounce Email',
	'NavBarName' => 'Ticket',
	'Title' => 'Bounce'
	};</pre>

6.20.30. Frontend::Module###AgentTicketForward

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AgentTicketForward'} = {
	'Description' => 'Ticket Forward Email',
	'Loader' => {
	'JavaScript' => [
	'Core.Agent.CustomerSearch.js',
	'Core.Agent.TicketAction.js'
	]
	},
	'NavBarName' => 'Ticket',
	'Title' => 'Forward'
	};</pre>

6.20.31. Frontend::Module###AgentTicketCustomer

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AgentTicketCustomer'} = {
	'Description' => 'Ticket Customer',
	'Loader' => {
	'JavaScript' => [
	'Core.Agent.CustomerSearch.js',
	'Core.Agent.TicketAction.js'
	]
	},
	'NavBarName' => 'Ticket',
	'Title' => 'Customer'
	};</pre>

6.20.32. Frontend::Module###AgentTicketClose

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AgentTicketClose'} = {
	'Description' => 'Ticket Close',
	'Loader' => {
	'JavaScript' => [
	'Core.Agent.TicketAction.js'
	]
	},
	'NavBarName' => 'Ticket',
	'Title' => 'Close'
	};</pre>

6.20.33. Frontend::Module###AgentTicketFreeText

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>$Self->{'Frontend::Module'}->{'AgentTicketFreeText'} = {</pre>

```
  'Description' => 'Ticket FreeText',
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.TicketAction.js'
    ]
  },
  'NavBarName' => 'Ticket',
  'Title' => 'Free Fields'
};
```

6.20.34. Frontend::Module###AgentTicketPrint

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketPrint'} = {`
	`'Description' => 'Ticket Print',`
	`'NavBarName' => 'Ticket',`
	`'Title' => 'Print'`
	`};`

6.20.35. Frontend::Module###AgentTicketBulk

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketBulk'} = {`
	`'Description' => 'Ticket bulk module',`
	`'Loader' => {`
	`'JavaScript' => [`
	`'Core.Agent.TicketAction.js'`
	`]`
	`},`
	`'NavBarName' => 'Ticket',`
	`'Title' => 'Bulk-Action'`
	`};`

6.21. Frontend::Agent::Preferences

6.21.1. PreferencesGroups###NewTicketNotify

Description:	Parameters for the NewTicketNotify object in the preferences view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	`$Self->{'PreferencesGroups'}->{'NewTicketNotify'} = {`
	`'Active' => '1',`
	`'Column' => 'Email Settings',`
	`'Data' => {`

```
    '0' => 'No',
    '1' => 'Yes'
  },
  'DataSelected' => '0',
  'Desc' => 'Send me a notification if there is a new ticket in "My
 Queues".',
  'Key' => 'Send new ticket notifications',
  'Label' => 'New ticket notification',
  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',
  'PrefKey' => 'UserSendNewTicketNotification',
  'Prio' => '1000'
};
```

6.21.2. PreferencesGroups###FollowUpNotify

Description:	Parameters for the FollowUpNotify object in the preference view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'PreferencesGroups'}->{'FollowUpNotify'} =  {
  'Active' => '1',
  'Column' => 'Email Settings',
  'Data' => {
    '0' => 'No',
    '1' => 'Yes'
  },
  'DataSelected' => '0',
  'Desc' => 'Send me a notification if a customer sends a follow up
 and I\'m the owner of the ticket or the ticket is unlocked and is
 in one of my subscribed queues.',
  'Key' => 'Send ticket follow up notifications',
  'Label' => 'Ticket follow up notification',
  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',
  'PrefKey' => 'UserSendFollowUpNotification',
  'Prio' => '2000'
};
```

6.21.3. PreferencesGroups###LockTimeoutNotify

Description:	Parameters for the LockTimeoutNotify object in the preference view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'PreferencesGroups'}->{'LockTimeoutNotify'} =  {
  'Active' => '1',
  'Column' => 'Email Settings',
  'Data' => {
    '0' => 'No',
    '1' => 'Yes'
  },
  'DataSelected' => '0',
  'Desc' => 'Send me a notification if a ticket is unlocked by the
 system.',
  'Key' => 'Send ticket lock timeout notifications',
  'Label' => 'Ticket lock timeout notification',
  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',
  'PrefKey' => 'UserSendLockTimeoutNotification',
  'Prio' => '3000'
```

	};

6.21.4. PreferencesGroups###MoveNotify

Description:	Parameters for the MoveNotify object in the preference view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	$Self->{'PreferencesGroups'}->{'MoveNotify'} = {
	'Active' => '1',
	'Column' => 'Email Settings',
	'Data' => {
	'0' => 'No',
	'1' => 'Yes'
	},
	'DataSelected' => '0',
	'Desc' => 'Send me a notification if a ticket is moved into one of "My Queues".',
	'Key' => 'Send ticket move notifications',
	'Label' => 'Ticket move notification',
	'Module' => 'Kernel::Output::HTML::PreferencesGeneric',
	'PrefKey' => 'UserSendMoveNotification',
	'Prio' => '4000'
	};

6.21.5. PreferencesGroups###WatcherNotify

Description:	Parameters for the WatcherNotify object in the preference view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	$Self->{'PreferencesGroups'}->{'WatcherNotify'} = {
	'Active' => '1',
	'Column' => 'Email Settings',
	'Data' => {
	'0' => 'No',
	'1' => 'Yes'
	},
	'DataSelected' => '0',
	'Desc' => 'Send me the same notifications for my watched tickets that the ticket owners will get.',
	'Key' => 'Send ticket watch notifications',
	'Label' => 'Ticket watch notification',
	'Module' => 'Kernel::Output::HTML::PreferencesTicketWatcher',
	'PrefKey' => 'UserSendWatcherNotification',
	'Prio' => '5000'
	};

6.21.6. PreferencesGroups###CustomQueue

Description:	Parameters for the CustomQueue object in the preference view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences

Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'PreferencesGroups'}->{'CustomQueue'} = {
	  'Active' => '1',
	  'Column' => 'Other Settings',
	  'Desc' => 'Your queue selection of your favorite queues. You also get notified about those queues via email if enabled.',
	  'Key' => 'My Queues',
	  'Label' => 'My Queues',
	  'Module' => 'Kernel::Output::HTML::PreferencesCustomQueue',
	  'Permission' => 'ro',
	  'Prio' => '1000'
	};</pre>

6.21.7. PreferencesGroups###RefreshTime

Description:	Parameters for the RefreshTime object in the preference view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'PreferencesGroups'}->{'RefreshTime'} = {
	  'Active' => '1',
	  'Column' => 'Other Settings',
	  'Data' => {
	    '0' => 'off',
	    '10' => '10 minutes',
	    '15' => '15 minutes',
	    '2' => ' 2 minutes',
	    '5' => ' 5 minutes',
	    '7' => ' 7 minutes'
	  },
	  'DataSelected' => '0',
	  'Desc' => 'If enabled, the different overviews (Dashboard, LockedView, QueueView) will automatically refresh after the specified time.',
	  'Key' => 'Refresh Overviews after',
	  'Label' => 'Overview Refresh Time',
	  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',
	  'PrefKey' => 'UserRefreshTime',
	  'Prio' => '2000'
	};</pre>

6.21.8. PreferencesGroups###TicketOverviewSmallPageShown

Description:	Parameters for the pages (in which the tickets are shown) of the small ticket overview.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'PreferencesGroups'}->{'TicketOverviewSmallPageShown'} = {
	  'Active' => '0',
	  'Column' => 'Other Settings',
	  'Data' => {
	    '10' => '10',
	    '15' => '15',</pre>

```
    '20' => '20',
    '25' => '25',
    '30' => '30',
    '35' => '35'
  },
  'DataSelected' => '25',
  'Key' => 'Ticket limit per page for Ticket Overview "Small"',
  'Label' => 'Ticket Overview "Small" Limit',
  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',
  'PrefKey' => 'UserTicketOverviewSmallPageShown',
  'Prio' => '8000'
};
```

6.21.9. PreferencesGroups###TicketOverviewFilterSettings

Description:	Parameters for .
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'PreferencesGroups'}->{'TicketOverviewFilterSettings'} =  {
  'Active' => '0',
  'Column' => 'Other Settings',
  'Key' => 'Column ticket filters for Ticket Overviews type
 "Small".',
  'Label' => 'Enabled filters.',
  'Module' => 'Kernel::Output::HTML::PreferencesColumnFilters',
  'PrefKey' => 'UserFilterColumnsEnabled',
  'Prio' => '8100'
};
```

6.21.10. PreferencesGroups###TicketOverviewMediumPageShown

Description:	Parameters for the pages (in which the tickets are shown) of the medium ticket overview.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'PreferencesGroups'}->{'TicketOverviewMediumPageShown'} =
{
  'Active' => '0',
  'Column' => 'Other Settings',
  'Data' => {
    '10' => '10',
    '15' => '15',
    '20' => '20',
    '25' => '25',
    '30' => '30',
    '35' => '35'
  },
  'DataSelected' => '20',
  'Key' => 'Ticket limit per page for Ticket Overview "Medium"',
  'Label' => 'Ticket Overview "Medium" Limit',
  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',
  'PrefKey' => 'UserTicketOverviewMediumPageShown',
  'Prio' => '8100'
};
```

6.21.11. PreferencesGroups###TicketOverviewPreviewPageShown

Description:	Parameters for the pages (in which the tickets are shown) of the ticket preview overview.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'PreferencesGroups'}->{'TicketOverviewPreviewPageShown'} = {
	'Active' => '0',
	'Column' => 'Other Settings',
	'Data' => {
	'10' => '10',
	'15' => '15',
	'20' => '20',
	'25' => '25',
	'30' => '30',
	'35' => '35'
	},
	'DataSelected' => '15',
	'Key' => 'Ticket limit per page for Ticket Overview "Preview"',
	'Label' => 'Ticket Overview "Preview" Limit',
	'Module' => 'Kernel::Output::HTML::PreferencesGeneric',
	'PrefKey' => 'UserTicketOverviewPreviewPageShown',
	'Prio' => '8200'
	};</pre>

6.21.12. PreferencesGroups###CreateNextMask

Description:	Parameters for the CreateNextMask object in the preference view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'PreferencesGroups'}->{'CreateNextMask'} = {
	'Active' => '1',
	'Column' => 'Other Settings',
	'Data' => {
	'' => 'CreateTicket',
	'AgentTicketZoom' => 'TicketZoom'
	},
	'DataSelected' => '',
	'Key' => 'Show this screen after I created a new ticket',
	'Label' => 'Screen after new ticket',
	'Module' => 'Kernel::Output::HTML::PreferencesGeneric',
	'PrefKey' => 'UserCreateNextMask',
	'Prio' => '3000'
	};</pre>

6.22. Frontend::Agent::SearchRouter

6.22.1. Frontend::Search###AgentCustomerInformationCenter

Description:	Search backend router.

Group:	Ticket
SubGroup:	Frontend::Agent::SearchRouter
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Search'}->{'AgentCustomerInformationCenter'} = { '^AgentCustomerInformationCenter' => 'Action=AgentCustomerInformationCenterSearch' };`

6.22.2. Frontend::Search###Ticket

Description:	Search backend router.
Group:	Ticket
SubGroup:	Frontend::Agent::SearchRouter
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Search'}->{'Ticket'} = { '^AgentTicket' => 'Action=AgentTicketSearch;Subaction=AJAX' };`

6.23. Frontend::Agent::Ticket::ArticleAttachmentModule

6.23.1. Ticket::Frontend::ArticleAttachmentModule###1-Download

Description:	Shows a link to download article attachments in the zoom view of the article in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleAttachmentModule
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ArticleAttachmentModule'}->{'1-Download'} = { 'Module' => 'Kernel::Output::HTML::ArticleAttachmentDownload' };`

6.23.2. Ticket::Frontend::ArticleAttachmentModule###2-HTML-Viewer

Description:	Shows a link to access article attachments via a html online viewer in the zoom view of the article in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleAttachmentModule
Valid:	1

Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ArticleAttachmentModule'}->{'2-HTML-Viewer'} = { 'Module' => 'Kernel::Output::HTML::ArticleAttachmentHTMLViewer' };`

6.24. Frontend::Agent::Ticket::ArticleComposeModule

6.24.1. Ticket::Frontend::ArticleComposeModule###1-SignEmail

Description:	Module to compose signed messages (PGP or S/MIME).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleComposeModule
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ArticleComposeModule'}->{'1-SignEmail'} = { 'Module' => 'Kernel::Output::HTML::ArticleComposeSign' };`

6.24.2. Ticket::Frontend::ArticleComposeModule###2-CryptEmail

Description:	Module to crypt composed messages (PGP or S/MIME).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleComposeModule
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ArticleComposeModule'}->{'2-CryptEmail'} = { 'Module' => 'Kernel::Output::HTML::ArticleComposeCrypt' };`

6.25. Frontend::Agent::Ticket::ArticleViewModule

6.25.1. Ticket::Frontend::ArticleViewModule###1-PGP

Description:	Agent interface article notification module to check PGP.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleViewModule
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ArticleViewModule'}->{'1-PGP'} = { 'Module' => 'Kernel::Output::HTML::ArticleCheckPGP' };`

6.25.2. Ticket::Frontend::ArticleViewModule###1-SMIME

Description:	Agent interface module to check incoming emails in the Ticket-Zoom-View if the S/MIME-key is available and true.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleViewModule
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ArticleViewModule'}->{'1-SMIME'} = { 'Module' => 'Kernel::Output::HTML::ArticleCheckSMIME' };`

6.26. Frontend::Agent::Ticket::ArticleViewModulePre

6.26.1. Ticket::Frontend::ArticlePreViewModule###1-PGP

Description:	Agent interface article notification module to check PGP.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleViewModulePre
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ArticlePreViewModule'}->{'1-PGP'} = { 'Module' => 'Kernel::Output::HTML::ArticleCheckPGP' };`

6.26.2. Ticket::Frontend::ArticlePreViewModule###1-SMIME

Description:	Agent interface article notification module to check S/MIME.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleViewModulePre
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ArticlePreViewModule'}->{'1-SMIME'} = { 'Module' => 'Kernel::Output::HTML::ArticleCheckSMIME' };`

6.27. Frontend::Agent::Ticket::MenuModule

6.27.1. Ticket::Frontend::MenuModule###000-Back

Description:	Shows a link in the menu to go back in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule

Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'000-Back'} = {`
	`'Action' => '',`
	`'Description' => 'Back',`
	`'Link' => '$Env{"LastScreenOverview"};TicketID=`
	`$QData{"TicketID"}',`
	`'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',`
	`'Name' => 'Back',`
	`'PopupType' => '',`
	`'Target' => ''`
	`};`

6.27.2. Ticket::Frontend::MenuModule###100-Lock

Description:	Shows a link in the menu to lock/unlock tickets in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'100-Lock'} = {`
	`'Action' => 'AgentTicketLock',`
	`'Module' => 'Kernel::Output::HTML::TicketMenuLock',`
	`'Name' => 'Lock',`
	`'Target' => ''`
	`};`

6.27.3. Ticket::Frontend::MenuModule###200-History

Description:	Shows a link in the menu to access the history of a ticket in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'200-History'} = {`
	`'Action' => 'AgentTicketHistory',`
	`'Description' => 'Show the ticket history',`
	`'Link' => 'Action=AgentTicketHistory;TicketID=$QData{"TicketID"}',`
	`'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',`
	`'Name' => 'History',`
	`'PopupType' => 'TicketHistory',`
	`'Target' => ''`
	`};`

6.27.4. Ticket::Frontend::MenuModule###210-Print

Description:	Shows a link in the menu to print a ticket or an article in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'210-Print'} = {`

```
  'Action' => 'AgentTicketPrint',
  'Description' => 'Print this ticket',
  'Link' => 'Action=AgentTicketPrint;TicketID=$QData{"TicketID"}',
  'LinkParam' => 'target="print"',
  'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',
  'Name' => 'Print',
  'PopupType' => 'TicketAction',
  'Target' => ''
};
```

6.27.5. Ticket::Frontend::MenuModule###300-Priority

Description:	Shows a link in the menu to see the priority of a ticket in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'300-Priority'} = { 'Action' => 'AgentTicketPriority', 'Description' => 'Change the ticket priority', 'Link' => 'Action=AgentTicketPriority;TicketID= $QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Priority', 'PopupType' => 'TicketAction', 'Target' => '' };`

6.27.6. Ticket::Frontend::MenuModule###310-FreeText

Description:	Shows a link in the menu to add a free text field in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'310-FreeText'} = { 'Action' => 'AgentTicketFreeText', 'Description' => 'Change the free fields for this ticket', 'Link' => 'Action=AgentTicketFreeText;TicketID= $QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Free Fields', 'PopupType' => 'TicketAction', 'Target' => '' };`

6.27.7. Ticket::Frontend::MenuModule###320-Link

Description:	Shows a link in the menu that allows linking a ticket with another object in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Ticket::Frontend::MenuModule'}->{'320-Link'} = {
	'Action' => 'AgentLinkObject',
	'Description' => 'Link this ticket to other objects',
	'Link' => 'Action=AgentLinkObject;SourceObject=Ticket;SourceKey=
	$QData{"TicketID"}',
	'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',
	'Name' => 'Link',
	'PopupType' => 'TicketAction',
	'Target' => ''
	};</pre>

6.27.8. Ticket::Frontend::MenuModule###400-Owner

Description:	Shows a link in the menu to see the owner of a ticket in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Ticket::Frontend::MenuModule'}->{'400-Owner'} = {
	'Action' => 'AgentTicketOwner',
	'Description' => 'Change the owner for this ticket',
	'Link' => 'Action=AgentTicketOwner;TicketID=$QData{"TicketID"}',
	'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',
	'Name' => 'Owner',
	'PopupType' => 'TicketAction',
	'Target' => ''
	};</pre>

6.27.9. Ticket::Frontend::MenuModule###410-Responsible

Description:	Shows a link in the menu to see the responsible agent of a ticket in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Ticket::Frontend::MenuModule'}->{'410-Responsible'} = {
	'Action' => 'AgentTicketResponsible',
	'Description' => 'Change the responsible person for this ticket',
	'Link' => 'Action=AgentTicketResponsible;TicketID=
	$QData{"TicketID"}',
	'Module' => 'Kernel::Output::HTML::TicketMenuResponsible',
	'Name' => 'Responsible',
	'PopupType' => 'TicketAction',
	'Target' => ''
	};</pre>

6.27.10. Ticket::Frontend::MenuModule###420-Customer

Description:	Shows a link in the menu to see the customer who requested the ticket in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule

Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Ticket::Frontend::MenuModule'}->{'420-Customer'} = {
	'Action' => 'AgentTicketCustomer',
	'Description' => 'Change the customer for this ticket',
	'Link' => 'Action=AgentTicketCustomer;TicketID=
	$QData{"TicketID"}',
	'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',
	'Name' => 'Customer',
	'PopupType' => 'TicketAction',
	'Target' => ''
	};</pre>

6.27.11. Ticket::Frontend::MenuModule###420-Note

Description:	Shows a link in the menu to add a note in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Ticket::Frontend::MenuModule'}->{'420-Note'} = {
	'Action' => 'AgentTicketNote',
	'Description' => 'Add a note to this ticket',
	'Link' => 'Action=AgentTicketNote;TicketID=$QData{"TicketID"}',
	'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',
	'Name' => 'Note',
	'PopupType' => 'TicketAction',
	'Target' => ''
	};</pre>

6.27.12. Ticket::Frontend::MenuModule###425-Phone Call Outbound

Description:	Shows a link in the menu to add a note in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Ticket::Frontend::MenuModule'}->{'425-Phone Call Outbound'} = {
	'Action' => 'AgentTicketPhoneOutbound',
	'Description' => 'Phone Call Outbound',
	'Link' => 'Action=AgentTicketPhoneOutbound;TicketID=
	$QData{"TicketID"}',
	'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',
	'Name' => 'Phone Call Outbound',
	'PopupType' => 'TicketAction',
	'Target' => ''
	};</pre>

6.27.13. Ticket::Frontend::MenuModule###426-Phone Call Inbound

Description:	Shows a link in the menu to add a note in the ticket zoom view of the agent interface.

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Ticket::Frontend::MenuModule'}->{'426-Phone Call Inbound'} = {
	'Action' => 'AgentTicketPhoneInbound',
	'Description' => 'Phone Call Inbound',
	'Link' => 'Action=AgentTicketPhoneInbound;TicketID=$QData{"TicketID"}',
	'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',
	'Name' => 'Phone Call Inbound',
	'PopupType' => 'TicketAction',
	'Target' => ''
	};</pre>

6.27.14. Ticket::Frontend::MenuModule###430-Merge

Description:	Shows a link in the menu that allows merging tickets in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Ticket::Frontend::MenuModule'}->{'430-Merge'} = {
	'Action' => 'AgentTicketMerge',
	'Description' => 'Merge into a different ticket',
	'Link' => 'Action=AgentTicketMerge;TicketID=$QData{"TicketID"}',
	'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',
	'Name' => 'Merge',
	'PopupType' => 'TicketAction',
	'Target' => ''
	};</pre>

6.27.15. Ticket::Frontend::MenuModule###440-Pending

Description:	Shows a link in the menu to set a ticket as pending in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Ticket::Frontend::MenuModule'}->{'440-Pending'} = {
	'Action' => 'AgentTicketPending',
	'Description' => 'Set this ticket to pending',
	'Link' => 'Action=AgentTicketPending;TicketID=$QData{"TicketID"}',
	'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',
	'Name' => 'Pending',
	'PopupType' => 'TicketAction',
	'Target' => ''
	};</pre>

6.27.16. Ticket::Frontend::MenuModule###448-Watch

Description:	Shows a link in the menu for subscribing / unsubscribing from a ticket in the ticket zoom view of the agent interface.

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Ticket::Frontend::MenuModule'}->{'448-Watch'} = {
	'Action' => 'AgentTicketWatcher',
	'Module' => 'Kernel::Output::HTML::TicketMenuTicketWatcher',
	'Name' => 'Watch',
	'Target' => ''
	};</pre>

6.27.17. Ticket::Frontend::MenuModule###450-Close

Description:	Shows a link in the menu to close a ticket in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Ticket::Frontend::MenuModule'}->{'450-Close'} = {
	'Action' => 'AgentTicketClose',
	'Description' => 'Close this ticket',
	'Link' => 'Action=AgentTicketClose;TicketID=$QData{"TicketID"}',
	'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',
	'Name' => 'Close',
	'PopupType' => 'TicketAction',
	'Target' => ''
	};</pre>

6.27.18. Ticket::Frontend::MenuModule###460-Delete

Description:	Shows a link in the menu to delete a ticket in the ticket zoom view of the agent interface. Additional access control to show or not show this link can be done by using Key "Group" and Content like "rw:group1;move_into:group2".
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	0
Required:	0

Config-Setting:	<pre>$Self->{'Ticket::Frontend::MenuModule'}->{'460-Delete'} = {
	'Action' => 'AgentTicketMove',
	'Description' => 'Delete this ticket',
	'Link' => 'Action=AgentTicketMove;TicketID=
	$Data{"TicketID"};DestQueue=Delete',
	'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',
	'Name' => 'Delete',
	'PopupType' => '',
	'Target' => ''
	};</pre>

6.27.19. Ticket::Frontend::MenuModule###470-Spam

Description:	Shows a link to set a ticket as spam in the ticket zoom view of the agent interface. Additional access control to show or not

	show this link can be done by using Key "Group" and Content like "rw:group1;move_into:group2".
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'470-Spam'} = {`
	`'Action' => 'AgentTicketMove',`
	`'Description' => 'Mark as Spam!',`
	`'Link' => 'Action=AgentTicketMove;TicketID=`
	`$Data{"TicketID"};DestQueue=Delete',`
	`'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',`
	`'Name' => 'Spam',`
	`'PopupType' => '',`
	`'Target' => ''`
	`};`

6.28. Frontend::Agent::Ticket::MenuModulePre

6.28.1. Ticket::Frontend::PreMenuModule###100-Lock

Description:	Shows a link in the menu to lock / unlock a ticket in the ticket overviews of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::PreMenuModule'}->{'100-Lock'} = {`
	`'Action' => 'AgentTicketLock',`
	`'Module' => 'Kernel::Output::HTML::TicketMenuLock',`
	`'Name' => 'Lock',`
	`'PopupType' => '',`
	`'Target' => ''`
	`};`

6.28.2. Ticket::Frontend::PreMenuModule###200-Zoom

Description:	Shows a link in the menu to zoom a ticket in the ticket overviews of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::PreMenuModule'}->{'200-Zoom'} = {`
	`'Action' => 'AgentTicketZoom',`
	`'Description' => 'Look into a ticket!',`
	`'Link' => 'Action=AgentTicketZoom;TicketID=$QData{"TicketID"}',`
	`'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',`
	`'Name' => 'Zoom',`
	`'PopupType' => '',`
	`'Target' => ''`
	`};`

6.28.3. Ticket::Frontend::PreMenuModule###210-History

Description:	Shows a link in the menu to see the history of a ticket in every ticket overview of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::PreMenuModule'}->{'210-History'} = {`
	`'Action' => 'AgentTicketHistory',`
	`'Description' => 'Show the ticket history',`
	`'Link' => 'Action=AgentTicketHistory;TicketID=$QData{"TicketID"}',`
	`'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',`
	`'Name' => 'History',`
	`'PopupType' => 'TicketHistory',`
	`'Target' => ''`
	`};`

6.28.4. Ticket::Frontend::PreMenuModule###300-Priority

Description:	Shows a link in the menu to set the priority of a ticket in every ticket overview of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::PreMenuModule'}->{'300-Priority'} = {`
	`'Action' => 'AgentTicketPriority',`
	`'Description' => 'Change the priority for this ticket',`
	`'Link' => 'Action=AgentTicketPriority;TicketID=`
	`$QData{"TicketID"}',`
	`'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',`
	`'Name' => 'Priority',`
	`'PopupType' => 'TicketAction',`
	`'Target' => ''`
	`};`

6.28.5. Ticket::Frontend::PreMenuModule###420-Note

Description:	Shows a link in the menu to add a note to a ticket in every ticket overview of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::PreMenuModule'}->{'420-Note'} = {`
	`'Action' => 'AgentTicketNote',`
	`'Description' => 'Add a note to this ticket',`
	`'Link' => 'Action=AgentTicketNote;TicketID=$QData{"TicketID"}',`
	`'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',`
	`'Name' => 'Note',`
	`'PopupType' => 'TicketAction',`
	`'Target' => ''`
	`};`

6.28.6. Ticket::Frontend::PreMenuModule###440-Close

Description:	Shows a link in the menu to close a ticket in every ticket overview of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::PreMenuModule'}->{'440-Close'} = {`
	`'Action' => 'AgentTicketClose',`
	`'Description' => 'Close this ticket',`
	`'Link' => 'Action=AgentTicketClose;TicketID=$QData{"TicketID"}',`
	`'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',`
	`'Name' => 'Close',`
	`'PopupType' => 'TicketAction',`
	`'Target' => ''`
	`};`

6.28.7. Ticket::Frontend::PreMenuModule###445-Move

Description:	Shows a link in the menu to move a ticket in every ticket overview of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::PreMenuModule'}->{'445-Move'} = {`
	`'Action' => 'AgentTicketMove',`
	`'Description' => 'Change queue!',`
	`'Module' => 'Kernel::Output::HTML::TicketMenuMove',`
	`'Name' => 'Move'`
	`};`

6.28.8. Ticket::Frontend::PreMenuModule###450-Delete

Description:	Shows a link in the menu to delete a ticket in every ticket overview of the agent interface. Additional access control to show or not show this link can be done by using Key "Group" and Content like "rw:group1;move_into:group2".
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::PreMenuModule'}->{'450-Delete'} = {`
	`'Action' => 'AgentTicketMove',`
	`'Description' => 'Delete this ticket',`
	`'Link' => 'Action=AgentTicketMove;TicketID=`
	`$Data{"TicketID"};DestQueue=Delete',`
	`'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',`
	`'Name' => 'Delete',`
	`'PopupType' => '',`
	`'Target' => ''`

	};

6.28.9. Ticket::Frontend::PreMenuModule###460-Spam

Description:	Shows a link in the menu to set a ticket as spam in every ticket overview of the agent interface. Additional access control to show or not show this link can be done by using Key "Group" and Content like "rw:group1;move_into:group2".
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::PreMenuModule'}->{'460-Spam'} = {`
	`'Action' => 'AgentTicketMove',`
	`'Description' => 'Mark as Spam!',`
	`'Link' => 'Action=AgentTicketMove;TicketID=`
	`$Data{"TicketID"};DestQueue=Delete',`
	`'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',`
	`'Name' => 'Spam',`
	`'PopupType' => '',`
	`'Target' => ''`
	`};`

6.29. Frontend::Agent::Ticket::OverviewMenuModule

6.29.1. Ticket::Frontend::OverviewMenuModule###001-Sort

Description:	Shows a select of ticket attributes to order the queue view ticket list. The possible selections can be configured via 'TicketOverviewMenuSort###SortAttributes'.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::OverviewMenuModule
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::OverviewMenuModule'}->{'001-Sort'} = {`
	`'Module' => 'Kernel::Output::HTML::TicketOverviewMenuSort'`
	`};`

6.29.2. TicketOverviewMenuSort###SortAttributes

Description:	Defines from which ticket attributes the agent can select the result order.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::OverviewMenuModule
Valid:	1
Required:	0

Config-Setting:	`$Self->{'TicketOverviewMenuSort'}->{'SortAttributes'} = {`
	`'Age' => '1',`
	`'Title' => '1'`
	`};`

6.30. Frontend::Agent::Ticket::ViewBounce

6.30.1. Ticket::Frontend::AgentTicketBounce###Permission

Description:	Required permissions to use the ticket bounce screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBounce
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBounce'}->{'Permission'} = 'bounce';`

6.30.2. Ticket::Frontend::AgentTicketBounce###RequiredLock

Description:	Defines if a ticket lock is required in the ticket bounce screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBounce
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBounce'}->{'RequiredLock'} = '1';`

6.30.3. Ticket::Frontend::AgentTicketBounce###StateDefault

Description:	Defines the default next state of a ticket after being bounced, in the ticket bounce screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBounce
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBounce'}->{'StateDefault'} = 'closed successful';`

6.30.4. Ticket::Frontend::AgentTicketBounce###StateType

Description:	Defines the next state of a ticket after being bounced, in the ticket bounce screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBounce
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBounce'}->{'StateType'} = [ 'open', 'closed' ];`

6.30.5. Ticket::Frontend::BounceText

Description:	Defines the default ticket bounced notification for customer/ sender in the ticket bounce screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBounce
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::BounceText'} = 'Your email with ticket number "<OTRS_TICKET>" is bounced to "<OTRS_BOUNCE_TO>". Contact this address for further information.';`

6.31. Frontend::Agent::Ticket::ViewBulk

6.31.1. Ticket::Frontend::AgentTicketBulk###RequiredLock

Description:	Automatically lock and set owner to current Agent after selecting for an Bulk Action.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'RequiredLock'} = '1';`

6.31.2. Ticket::Frontend::AgentTicketBulk###TicketType

Description:	Sets the ticket type in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'TicketType'} = '1';`

6.31.3. Ticket::Frontend::AgentTicketBulk###Owner

Description:	Sets the ticket owner in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'Owner'} = '1';`

6.31.4. Ticket::Frontend::AgentTicketBulk###Responsible

Description:	Sets the responsible agent of the ticket in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'Responsible'} = '1';`

6.31.5. Ticket::Frontend::AgentTicketBulk###State

Description:	If a note is added by an agent, sets the state of a ticket in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'State'} = '1';`

6.31.6. Ticket::Frontend::AgentTicketBulk###StateType

Description:	Defines the next state of a ticket after adding a note, in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'StateType'} = [ 'open', 'closed', 'pending reminder', 'pending auto' ];`

6.31.7. Ticket::Frontend::AgentTicketBulk###StateDefault

Description:	Defines the default next state of a ticket after adding a note, in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'StateDefault'} = 'open';`

6.31.8. Ticket::Frontend::AgentTicketBulk###Priority

Description:	Shows the ticket priority options in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'Priority'} = '1';`

6.31.9. Ticket::Frontend::AgentTicketBulk###PriorityDefault

Description:	Defines the default ticket priority in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'PriorityDefault'} = '3 normal';`

6.31.10. Ticket::Frontend::AgentTicketBulk###ArticleTypeDefault

Description:	Defines the default type of the note in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'ArticleTypeDefault'} = 'note-internal';`

6.31.11. Ticket::Frontend::AgentTicketBulk###ArticleTypes

Description:	Specifies the different note types that will be used in the system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'ArticleTypes'} = { 'note-external' => '1', 'note-internal' => '1', 'note-report' => '0'`

	`};`

6.32. Frontend::Agent::Ticket::ViewClose

6.32.1. Ticket::Frontend::AgentTicketClose###Permission

Description:	Required permissions to use the close ticket screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Permission'} = 'close';`

6.32.2. Ticket::Frontend::AgentTicketClose###RequiredLock

Description:	Defines if a ticket lock is required in the close ticket screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'RequiredLock'} = '1';`

6.32.3. Ticket::Frontend::AgentTicketClose###TicketType

Description:	Sets the ticket type in the close ticket screen of the agent interface (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'TicketType'} = '0';`

6.32.4. Ticket::Frontend::AgentTicketClose###Service

Description:	Sets the service in the close ticket screen of the agent interface (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Service'} = '0';`

6.32.5. Ticket::Frontend::AgentTicketClose###ServiceMandatory

Description:	Sets if service must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'ServiceMandatory'} = '0';`

6.32.6. Ticket::Frontend::AgentTicketClose###SLAMandatory

Description:	Sets if SLA must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'SLAMandatory'} = '0';`

6.32.7. Ticket::Frontend::AgentTicketClose###Queue

Description:	Sets the queue in the ticket close screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Queue'} = '0';`

6.32.8. Ticket::Frontend::AgentTicketClose###Owner

Description:	Sets the ticket owner in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Owner'} = '0';`

6.32.9. Ticket::Frontend::AgentTicketClose###OwnerMandatory

Description:	Sets if ticket owner must be selected by the agent.

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'OwnerMandatory'} = '0';`

6.32.10. Ticket::Frontend::AgentTicketClose###Responsible

Description:	Sets the responsible agent of the ticket in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Responsible'} = '0';`

6.32.11. Ticket::Frontend::AgentTicketClose###State

Description:	If a note is added by an agent, sets the state of a ticket in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'State'} = '1';`

6.32.12. Ticket::Frontend::AgentTicketClose###StateType

Description:	Defines the next state of a ticket after adding a note, in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'StateType'} = [ 'closed' ];`

6.32.13. Ticket::Frontend::AgentTicketClose###StateDefault

Description:	Defines the default next state of a ticket after adding a note, in the close ticket screen of the agent interface.
Group:	Ticket

SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'StateDefault'} = 'closed successful';`

6.32.14. Ticket::Frontend::AgentTicketClose###Note

Description:	Allows adding notes in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Note'} = '1';`

6.32.15. Ticket::Frontend::AgentTicketClose###NoteMandatory

Description:	Sets if note must be filled in by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'NoteMandatory'} = '1';`

6.32.16. Ticket::Frontend::AgentTicketClose###Subject

Description:	Sets the default subject for notes added in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Subject'} = '$Text{"Close"}';`

6.32.17. Ticket::Frontend::AgentTicketClose###Body

Description:	Sets the default body text for notes added in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Body'} = '';`

6.32.18. Ticket::Frontend::AgentTicketClose###InvolvedAgent

Description:	Shows a list of all the involved agents on this ticket, in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'InvolvedAgent'} = '0';`

6.32.19. Ticket::Frontend::AgentTicketClose###InformAgent

Description:	Shows a list of all the possible agents (all agents with note permissions on the queue/ticket) to determine who should be informed about this note, in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'InformAgent'} = '0';`

6.32.20. Ticket::Frontend::AgentTicketClose###ArticleTypeDefault

Description:	Defines the default type of the note in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'ArticleTypeDefault'} = 'note-internal';`

6.32.21. Ticket::Frontend::AgentTicketClose###ArticleTypes

Description:	Specifies the available note types for this ticket mask. If the option is deselected, ArticleTypeDefault is used and the option is removed from the mask.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'ArticleTypes'} = { 'note-external' => '0', 'note-internal' => '1', 'note-report' => '0' };`

6.32.22. Ticket::Frontend::AgentTicketClose###Priority

Description:	Shows the ticket priority options in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Priority'} = '0';`

6.32.23. Ticket::Frontend::AgentTicketClose###PriorityDefault

Description:	Defines the default ticket priority in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'PriorityDefault'} = '3 normal';`

6.32.24. Ticket::Frontend::AgentTicketClose###Title

Description:	Shows the title fields in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Title'} = '0';`

6.32.25. Ticket::Frontend::AgentTicketClose###HistoryType

Description:	Defines the history type for the close ticket screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'HistoryType'} = 'AddNote';`

6.32.26. Ticket::Frontend::AgentTicketClose###HistoryComment

Description:	Defines the history comment for the close ticket screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'HistoryComment'} = '%%Close';`

6.32.27. Ticket::Frontend::AgentTicketClose###DynamicField

Description:	Dynamic fields shown in the ticket close screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'DynamicField'} = {};`

6.32.28. Ticket::Frontend::AgentTicketClose###RichTextWidth

Description:	Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'RichTextWidth'} = '620';`

6.32.29. Ticket::Frontend::AgentTicketClose###RichTextHeight

Description:	Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1

Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'RichTextHeight'} = '100';`

6.33. Frontend::Agent::Ticket::ViewCompose

6.33.1. Ticket::Frontend::AgentTicketCompose###Permission

Description:	Required permissions to use the ticket compose screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'Permission'} = 'compose';`

6.33.2. Ticket::Frontend::AgentTicketCompose###RequiredLock

Description:	Defines if a ticket lock is required in the ticket compose screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'RequiredLock'} = '1';`

6.33.3. Ticket::Frontend::AgentTicketCompose###StateDefault

Description:	Defines the default next state of a ticket if it is composed / answered in the ticket compose screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'StateDefault'} = 'open';`

6.33.4. Ticket::Frontend::AgentTicketCompose###StateType

Description:	Defines the next possible states after composing / answering a ticket in the ticket compose screen of the agent interface.

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'StateType'} = [ 'open', 'closed', 'pending auto', 'pending reminder' ];`

6.33.5. Ticket::Frontend::AgentTicketCompose###ArticleTypes

Description:	Specifies the different article types that will be used in the system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'ArticleTypes'} = [ 'email-external', 'email-internal' ];`

6.33.6. Ticket::Frontend::AgentTicketCompose###DefaultArticleType

Description:	Specifies the default article type for the ticket compose screen in the agent interface if the article type cannot be automatically detected.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'DefaultArticleType'} = 'email-external';`

6.33.7. Ticket::Frontend::ResponseFormat

Description:	Defines the format of responses in the ticket compose screen of the agent interface ($QData{"OrigFrom"} is From 1:1, $QData{"OrigFromName"} is only realname of From).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ResponseFormat'} = '$QData{"Salutation"} $QData{"StdResponse"} $QData{"Signature"}`

	$TimeShort{"$QData{"Created"}"} - $QData{"OrigFromName"} $Text{"wrote"}: $QData{"Body"} ';

6.33.8. Ticket::Frontend::Quote

Description:	Defines the used character for plaintext email quotes in the ticket compose screen of the agent interface. If this is empty or inactive, original emails will not be quoted but appended to the response.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	0
Config-Setting:	$Self->{'Ticket::Frontend::Quote'} = '>';

6.33.9. Ticket::Frontend::ResponseQuoteMaxLines

Description:	Defines the maximum number of quoted lines to be added to responses.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	0
Required:	0
Config-Setting:	$Self->{'Ticket::Frontend::ResponseQuoteMaxLines'} = '99';

6.33.10. Ticket::Frontend::ComposeAddCustomerAddress

Description:	Adds customers email addresses to recipients in the ticket compose screen of the agent interface. The customers email address won't be added if the article type is email-internal.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	$Self->{'Ticket::Frontend::ComposeAddCustomerAddress'} = '1';

6.33.11. Ticket::Frontend::ComposeReplaceSenderAddress

Description:	Replaces the original sender with current customer's email address on compose answer in the ticket compose screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	$Self->{'Ticket::Frontend::ComposeReplaceSenderAddress'} = '0';

6.33.12. Ticket::Frontend::ComposeExcludeCcRecipients

Description:	Uses Cc recipients in reply Cc list on compose an email answer in the ticket compose screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ComposeExcludeCcRecipients'} = '0';`

6.33.13. Ticket::Frontend::AgentTicketCompose###DynamicField

Description:	Dynamic fields shown in the ticket compose screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'DynamicField'} = {};`

6.34. Frontend::Agent::Ticket::ViewCustomer

6.34.1. Ticket::Frontend::AgentTicketCustomer###Permission

Description:	Required permissions to change the customer of a ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCustomer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketCustomer'}->{'Permission'} = 'customer';`

6.34.2. Ticket::Frontend::AgentTicketCustomer###RequiredLock

Description:	Defines if a ticket lock is required to change the customer of a ticket in the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCustomer
Valid:	1

Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketCustomer'}->{'RequiredLock'} = '0';`

6.35. Frontend::Agent::Ticket::ViewEmailNew

6.35.1. Ticket::Frontend::AgentTicketEmail###Priority

Description:	Sets the default priority for new email tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'Priority'} = '3 normal';`

6.35.2. Ticket::Frontend::AgentTicketEmail###ArticleType

Description:	Sets the default article type for new email tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'ArticleType'} = 'email-external';`

6.35.3. Ticket::Frontend::AgentTicketEmail###SenderType

Description:	Sets the default sender type for new email tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'SenderType'} = 'agent';`

6.35.4. Ticket::Frontend::AgentTicketEmail###Subject

Description:	Sets the default subject for new email tickets (e.g. 'email Outbound') in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1

Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'Subject'} =  '';`

6.35.5. Ticket::Frontend::AgentTicketEmail###Body

Description:	Sets the default text for new email tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'Body'} =  '';`

6.35.6. Ticket::Frontend::AgentTicketEmail###StateDefault

Description:	Sets the default next ticket state, after the creation of an email ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'StateDefault'} = 'open';`

6.35.7. Ticket::Frontend::AgentTicketEmail###StateType

Description:	Determines the next possible ticket states, after the creation of a new email ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'StateType'} = [ 'open', 'pending auto', 'pending reminder', 'closed' ];`

6.35.8. Ticket::Frontend::AgentTicketEmail###HistoryType

Description:	Defines the history type for the email ticket screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1

Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'HistoryType'} = 'EmailAgent';`

6.35.9. Ticket::Frontend::AgentTicketEmail###HistoryComment

Description:	Defines the history comment for the email ticket screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'HistoryComment'} = '';`

6.35.10. Ticket::Frontend::AgentTicketEmail###ServiceMandatory

Description:	Sets if service must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'ServiceMandatory'} = '0';`

6.35.11. Ticket::Frontend::AgentTicketEmail###SLAMandatory

Description:	Sets if SLA must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'SLAMandatory'} = '0';`

6.35.12. Ticket::Frontend::AgentTicketEmail###DynamicField

Description:	Dynamic fields shown in the ticket email screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'DynamicField'} = {};`

6.35.13. Ticket::Frontend::AgentTicketEmail###RichTextWidth

Description:	Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'RichTextWidth'} = '620';`

6.35.14. Ticket::Frontend::AgentTicketEmail###RichTextHeight

Description:	Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'RichTextHeight'} = '320';`

6.36. Frontend::Agent::Ticket::ViewEscalation

6.36.1. Ticket::Frontend::AgentTicketEscalationView###TicketPermissio

Description:	Defines the required permission to show a ticket in the escalation view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEscalation
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEscalationView'}->{'TicketPermission'} = 'rw';`

6.36.2. Ticket::Frontend::AgentTicketEscalationView###ViewableTicket

Description:	Shows all open tickets (even if they are locked) in the escalation view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEscalation

Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEscalationView'}->{'ViewableTicketsPage'} = '50';`

6.36.3. Ticket::Frontend::AgentTicketEscalationView###SortBy::Default

Description:	Defines the default ticket attribute for ticket sorting in the escalation view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEscalation
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEscalationView'}->{'SortBy::Default'} = 'EscalationTime';`

6.36.4. Ticket::Frontend::AgentTicketEscalationView###Order::Default

Description:	Defines the default ticket order (after priority sort) in the escalation view of the agent interface. Up: oldest on top. Down: latest on top.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEscalation
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEscalationView'}->{'Order::Default'} = 'Up';`

6.36.5. Ticket::Frontend::AgentTicketEscalationView###DefaultColumns

Description:	Columns that can be filtered in the escalation view of the agent interface. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEscalation
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEscalationView'}->{'DefaultColumns'} = { 'Age' => '2', 'Changed' => '1', 'CustomerID' => '2', 'CustomerName' => '1', 'CustomerUserID' => '1', 'EscalationResponseTime' => '1', 'EscalationSolutionTime' => '1', 'EscalationTime' => '2', 'EscalationUpdateTime' => '1',`

	<pre> 'Lock' => '2',
	  'Owner' => '2',
	  'PendingTime' => '1',
	  'Priority' => '1',
	  'Queue' => '2',
	  'Responsible' => '1',
	  'SLA' => '1',
	  'Service' => '1',
	  'State' => '2',
	  'TicketNumber' => '2',
	  'Title' => '2',
	  'Type' => '1'
	};</pre>

6.37. Frontend::Agent::Ticket::ViewForward

6.37.1. Ticket::Frontend::AgentTicketForward###Permission

Description:	Required permissions to use the ticket forward screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	1
Config-Setting:	<pre>$Self->{'Ticket::Frontend::AgentTicketForward'}->{'Permission'} = 'forward';</pre>

6.37.2. Ticket::Frontend::AgentTicketForward###RequiredLock

Description:	Defines if a ticket lock is required in the ticket forward screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0
Config-Setting:	<pre>$Self->{'Ticket::Frontend::AgentTicketForward'}->{'RequiredLock'} = '1';</pre>

6.37.3. Ticket::Frontend::AgentTicketForward###StateDefault

Description:	Defines the default next state of a ticket after being forwarded, in the ticket forward screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0
Config-Setting:	<pre>$Self->{'Ticket::Frontend::AgentTicketForward'}->{'StateDefault'} = 'closed successful';</pre>

6.37.4. Ticket::Frontend::AgentTicketForward###StateType

Description:	Defines the next possible states after forwarding a ticket in the ticket forward screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketForward'}->{'StateType'} = [ 'open', 'closed', 'pending reminder', 'pending auto' ];`

6.37.5. Ticket::Frontend::AgentTicketForward###ArticleTypeDefault

Description:	Defines the default type of forwarded message in the ticket forward screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketForward'}->{'ArticleTypeDefault'} = 'email-external';`

6.37.6. Ticket::Frontend::AgentTicketForward###ArticleTypes

Description:	Specifies the different article types that will be used in the system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketForward'}->{'ArticleTypes'} = [ 'email-external', 'email-internal' ];`

6.37.7. Ticket::Frontend::AgentTicketForward###DynamicField

Description:	Dynamic fields shown in the ticket forward screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1

Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketForward'}->{'DynamicField'} = {};`

6.37.8. Ticket::Frontend::AgentTicketForward###RichTextWidth

Description:	Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketForward'}->{'RichTextWidth'} = '620';`

6.37.9. Ticket::Frontend::AgentTicketForward###RichTextHeight

Description:	Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketForward'}->{'RichTextHeight'} = '100';`

6.38. Frontend::Agent::Ticket::ViewFreeText

6.38.1. Ticket::Frontend::AgentTicketFreeText###Permission

Description:	Required permissions to use the ticket free text screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Permission'} = 'rw';`

6.38.2. Ticket::Frontend::AgentTicketFreeText###RequiredLock

Description:	Defines if a ticket lock is required in the ticket free text screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'RequiredLock'} = '0';`

6.38.3. Ticket::Frontend::AgentTicketFreeText###TicketType

Description:	Sets the ticket type in the ticket free text screen of the agent interface (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'TicketType'} = '1';`

6.38.4. Ticket::Frontend::AgentTicketFreeText###Service

Description:	Sets the service in the ticket free text screen of the agent interface (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Service'} = '1';`

6.38.5. Ticket::Frontend::AgentTicketFreeText###ServiceMandatory

Description:	Sets if service must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'ServiceMandatory'} = '0';`

6.38.6. Ticket::Frontend::AgentTicketFreeText###SLAMandatory

Description:	Sets if SLA must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText

Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'SLAMandatory'} = '0';`

6.38.7. Ticket::Frontend::AgentTicketFreeText###Queue

Description:	Sets the queue in the ticket free text screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Queue'} = '0';`

6.38.8. Ticket::Frontend::AgentTicketFreeText###Owner

Description:	Sets the ticket owner in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Owner'} = '0';`

6.38.9. Ticket::Frontend::AgentTicketFreeText###OwnerMandatory

Description:	Sets if ticket owner must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'OwnerMandatory'} = '0';`

6.38.10. Ticket::Frontend::AgentTicketFreeText###Responsible

Description:	Sets the responsible agent of the ticket in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Responsible'} = '0';`

6.38.11. Ticket::Frontend::AgentTicketFreeText###State

Description:	If a note is added by an agent, sets the state of a ticket in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'State'} = '0';`

6.38.12. Ticket::Frontend::AgentTicketFreeText###StateType

Description:	Defines the next state of a ticket after adding a note, in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'StateType'} = [ 'open', 'closed', 'pending reminder', 'pending auto' ];`

6.38.13. Ticket::Frontend::AgentTicketFreeText###StateDefault

Description:	Defines the default next state of a ticket after adding a note, in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'StateDefault'} = 'open';`

6.38.14. Ticket::Frontend::AgentTicketFreeText###Note

Description:	Allows adding notes in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText

Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Note'} = '0';`

6.38.15. Ticket::Frontend::AgentTicketFreeText###NoteMandatory

Description:	Sets if note must be filled in by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'NoteMandatory'} = '0';`

6.38.16. Ticket::Frontend::AgentTicketFreeText###Subject

Description:	Defines the default subject of a note in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Subject'} = '$Text{"Note"}';`

6.38.17. Ticket::Frontend::AgentTicketFreeText###Body

Description:	Defines the default body of a note in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Body'} = '';`

6.38.18. Ticket::Frontend::AgentTicketFreeText###InvolvedAgent

Description:	Shows a list of all the involved agents on this ticket, in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'InvolvedAgent'} = '0';`

6.38.19. Ticket::Frontend::AgentTicketFreeText###InformAgent

Description:	Shows a list of all the possible agents (all agents with note permissions on the queue/ticket) to determine who should be informed about this note, in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'InformAgent'} = '0';`

6.38.20. Ticket::Frontend::AgentTicketFreeText###ArticleTypeDefault

Description:	Defines the default type of the note in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'ArticleTypeDefault'} = 'note-internal';`

6.38.21. Ticket::Frontend::AgentTicketFreeText###ArticleTypes

Description:	Specifies the available note types for this ticket mask. If the option is deselected, ArticleTypeDefault is used and the option is removed from the mask.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'ArticleTypes'} = { 'note-external' => '1', 'note-internal' => '1', 'note-report' => '0' };`

6.38.22. Ticket::Frontend::AgentTicketFreeText###Priority

Description:	Shows the ticket priority options in the ticket free text screen of the agent interface.

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Priority'} = '0';`

6.38.23. Ticket::Frontend::AgentTicketFreeText###PriorityDefault

Description:	Defines the default ticket priority in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'PriorityDefault'} = '3 normal';`

6.38.24. Ticket::Frontend::AgentTicketFreeText###Title

Description:	Shows the title fields in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Title'} = '1';`

6.38.25. Ticket::Frontend::AgentTicketFreeText###HistoryType

Description:	Defines the history type for the ticket free text screen action, which gets used for ticket history.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'HistoryType'} = 'AddNote';`

6.38.26. Ticket::Frontend::AgentTicketFreeText###HistoryComment

Description:	Defines the history comment for the ticket free text screen action, which gets used for ticket history.
Group:	Ticket

SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'HistoryComment'} = '%%FreeText';`

6.38.27. Ticket::Frontend::AgentTicketFreeText###DynamicField

Description:	Dynamic fields shown in the ticket free text screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'DynamicField'} = {};`

6.38.28. Ticket::Frontend::AgentTicketFreeText###RichTextWidth

Description:	Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'RichTextWidth'} = '620';`

6.38.29. Ticket::Frontend::AgentTicketFreeText###RichTextHeight

Description:	Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'RichTextHeight'} = '100';`

6.39. Frontend::Agent::Ticket::ViewHistory

6.39.1. Ticket::Frontend::HistoryOrder

Description:	Shows the ticket history (reverse ordered) in the agent interface.
Group:	Ticket

SubGroup:	Frontend::Agent::Ticket::ViewHistory
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::HistoryOrder'} = 'normal';`

6.40. Frontend::Agent::Ticket::ViewLocked

6.40.1. Ticket::Frontend::AgentTicketLockedView###SortBy::Default

Description:	Defines the default ticket attribute for ticket sorting in the locked ticket view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewLocked
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketLockedView'}->{'SortBy::Default'} = 'Age';`

6.40.2. Ticket::Frontend::AgentTicketLockedView###Order::Default

Description:	Defines the default ticket order in the ticket locked view of the agent interface. Up: oldest on top. Down: latest on top.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewLocked
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketLockedView'}->{'Order::Default'} = 'Up';`

6.40.3. Ticket::Frontend::AgentTicketLockedView###DefaultColumns

Description:	Columns that can be filtered in the locked view of the agent interface. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewLocked
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketLockedView'}->{'DefaultColumns'} = { 'Age' => '2', 'Changed' => '1', 'CustomerID' => '2', 'CustomerName' => '1', 'CustomerUserID' => '1', 'EscalationResponseTime' => '1',`

```
  'EscalationSolutionTime' => '1',
  'EscalationTime' => '1',
  'EscalationUpdateTime' => '1',
  'Lock' => '2',
  'Owner' => '2',
  'PendingTime' => '1',
  'Priority' => '1',
  'Queue' => '2',
  'Responsible' => '1',
  'SLA' => '1',
  'Service' => '1',
  'State' => '2',
  'TicketNumber' => '2',
  'Title' => '2',
  'Type' => '1'
};
```

6.41. Frontend::Agent::Ticket::ViewMerge

6.41.1. Ticket::Frontend::AgentTicketMerge###Permission

Description:	Required permissions to use the ticket merge screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMerge
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMerge'}->{'Permission'} = 'rw';`

6.41.2. Ticket::Frontend::AgentTicketMerge###RequiredLock

Description:	Defines if a ticket lock is required in the ticket merge screen of a zoomed ticket in the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMerge
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMerge'}->{'RequiredLock'} = '1';`

6.41.3. Ticket::Frontend::MergeText

Description:	When tickets are merged, the customer can be informed per email by setting the check box "Inform Sender". In this text area, you can define a pre-formatted text which can later be modified by the agents.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMerge
Valid:	1

Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::MergeText'} = 'Your email with ticket number "<OTRS_TICKET>" is merged to "<OTRS_MERGE_TO_TICKET>".';`

6.41.4. Ticket::Frontend::AutomaticMergeSubject

Description:	When tickets are merged, a note will be added automatically to the ticket which is no longer active. Here you can define the subject of this note (this subject cannot be changed by the agent).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMerge
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AutomaticMergeSubject'} = 'Ticket Merged';`

6.41.5. Ticket::Frontend::AutomaticMergeText

Description:	When tickets are merged, a note will be added automatically to the ticket which is no longer active. Here you can define the body of this note (this text cannot be changed by the agent).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMerge
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AutomaticMergeText'} = 'Merged Ticket <OTRS_TICKET> to <OTRS_MERGE_TO_TICKET>.';`

6.41.6. Ticket::Frontend::AgentTicketMerge###RichTextWidth

Description:	Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMerge
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMerge'}->{'RichTextWidth'} = '620';`

6.41.7. Ticket::Frontend::AgentTicketMerge###RichTextHeight

Description:	Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMerge
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMerge'}->{'RichTextHeight'} = '100';`

6.42. Frontend::Agent::Ticket::ViewMove

6.42.1. Ticket::Frontend::MoveType

Description:	Determines if the list of possible queues to move to ticket into should be displayed in a dropdown list or in a new window in the agent interface. If "New Window" is set you can add a move note to the ticket.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::MoveType'} = 'form';`

6.42.2. Ticket::Frontend::AgentTicketMove###State

Description:	Allows to set a new ticket state in the move ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMove'}->{'State'} = '1';`

6.42.3. Ticket::Frontend::AgentTicketMove###StateType

Description:	Defines the next state of a ticket after being moved to another queue, in the move ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMove'}->{'StateType'} = [ 'open', 'closed' ];`

6.42.4. Ticket::Frontend::AgentTicketMove###Priority

Description:	Shows the ticket priority options in the move ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMove'}->{'Priority'} =  '0';`

6.42.5. Ticket::Frontend::AgentTicketMove###RequiredLock

Description:	Automatically lock and set owner to current Agent after opening the move ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMove'}->{'RequiredLock'} = '1';`

6.42.6. Ticket::Frontend::AgentTicketMove###NextScreen

Description:	Determines the next screen after the ticket is moved. LastScreenOverview will return the last overview screen (e.g. search results, queueview, dashboard). TicketZoom will return to the TicketZoom.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMove'}->{'NextScreen'} = 'TicketZoom';`

6.42.7. Ticket::Frontend::AgentTicketMove###Subject

Description:	Sets the default subject for notes added in the ticket move screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMove'}->{'Subject'} = '$Text{"Change Queue"}';`

6.42.8. Ticket::Frontend::AgentTicketMove###Body

Description:	Sets the default body text for notes added in the ticket move screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMove'}->{'Body'} =  '';`

6.42.9. Ticket::Frontend::AgentTicketMove###RichTextWidth

Description:	Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMove'}->{'RichTextWidth'} = '620';`

6.42.10. Ticket::Frontend::AgentTicketMove###RichTextHeight

Description:	Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMove'}->{'RichTextHeight'} = '100';`

6.42.11. Ticket::Frontend::AgentTicketMove###DynamicField

Description:	Dynamic fields shown in the ticket move screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMove'}->{'DynamicField'} = {};`

6.43. Frontend::Agent::Ticket::ViewNote

6.43.1. Ticket::Frontend::AgentTicketNote###Permission

Description:	Required permissions to use the ticket note screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Permission'} = 'note';`

6.43.2. Ticket::Frontend::AgentTicketNote###RequiredLock

Description:	Defines if a ticket lock is required in the ticket note screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'RequiredLock'} = '0';`

6.43.3. Ticket::Frontend::AgentTicketNote###TicketType

Description:	Sets the ticket type in the ticket note screen of the agent interface (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'TicketType'} = '0';`

6.43.4. Ticket::Frontend::AgentTicketNote###Service

Description:	Sets the service in the ticket note screen of the agent interface (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Service'} = '0';`

6.43.5. Ticket::Frontend::AgentTicketNote###ServiceMandatory

Description:	Sets if service must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'ServiceMandatory'} = '0';`

6.43.6. Ticket::Frontend::AgentTicketNote###SLAMandatory

Description:	Sets if SLA must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'SLAMandatory'} = '0';`

6.43.7. Ticket::Frontend::AgentTicketNote###Queue

Description:	Sets the queue in the ticket note screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Queue'} =  '0';`

6.43.8. Ticket::Frontend::AgentTicketNote###Owner

Description:	Sets the ticket owner in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Owner'} =  '0';`

6.43.9. Ticket::Frontend::AgentTicketNote###OwnerMandatory

Description:	Sets if ticket owner must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'OwnerMandatory'} = '0';`

6.43.10. Ticket::Frontend::AgentTicketNote###Responsible

Description:	Sets the responsible agent of the ticket in the ticket note screen of the agent interface.

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Responsible'} = '0';`

6.43.11. Ticket::Frontend::AgentTicketNote###State

Description:	If a note is added by an agent, sets the state of a ticket in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'State'} = '0';`

6.43.12. Ticket::Frontend::AgentTicketNote###StateType

Description:	Defines the next state of a ticket after adding a note, in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'StateType'} = [ 'open', 'closed', 'pending reminder', 'pending auto' ];`

6.43.13. Ticket::Frontend::AgentTicketNote###StateDefault

Description:	Defines the default next state of a ticket after adding a note, in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'StateDefault'} = 'open';`

6.43.14. Ticket::Frontend::AgentTicketNote###Note

Description:	Allows adding notes in the ticket note screen of the agent interface.

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Note'} = '1';`

6.43.15. Ticket::Frontend::AgentTicketNote###NoteMandatory

Description:	Sets if note must be filled in by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'NoteMandatory'} = '1';`

6.43.16. Ticket::Frontend::AgentTicketNote###Subject

Description:	Sets the default subject for notes added in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Subject'} = '$Text{"Note"}';`

6.43.17. Ticket::Frontend::AgentTicketNote###Body

Description:	Sets the default body text for notes added in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Body'} = '';`

6.43.18. Ticket::Frontend::AgentTicketNote###InvolvedAgent

Description:	Shows a list of all the involved agents on this ticket, in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'InvolvedAgent'} = '0';`

6.43.19. Ticket::Frontend::AgentTicketNote###InformAgent

Description:	Shows a list of all the possible agents (all agents with note permissions on the queue/ticket) to determine who should be informed about this note, in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'InformAgent'} = '0';`

6.43.20. Ticket::Frontend::AgentTicketNote###ArticleTypeDefault

Description:	Defines the default type of the note in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'ArticleTypeDefault'} = 'note-internal';`

6.43.21. Ticket::Frontend::AgentTicketNote###ArticleTypes

Description:	Specifies the available note types for this ticket mask. If the option is deselected, ArticleTypeDefault is used and the option is removed from the mask.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'ArticleTypes'} = { 'note-external' => '1', 'note-internal' => '1', 'note-report' => '0' };`

6.43.22. Ticket::Frontend::AgentTicketNote###Priority

Description:	Shows the ticket priority options in the ticket note screen of the agent interface.
Group:	Ticket

SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Priority'} = '0';`

6.43.23. Ticket::Frontend::AgentTicketNote###PriorityDefault

Description:	Defines the default ticket priority in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'PriorityDefault'} = '3 normal';`

6.43.24. Ticket::Frontend::AgentTicketNote###Title

Description:	Shows the title fields in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Title'} = '0';`

6.43.25. Ticket::Frontend::AgentTicketNote###HistoryType

Description:	Defines the history type for the ticket note screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'HistoryType'} = 'AddNote';`

6.43.26. Ticket::Frontend::AgentTicketNote###HistoryComment

Description:	Defines the history comment for the ticket note screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1

Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'HistoryComment'} = '%%Note';`

6.43.27. Ticket::Frontend::AgentTicketNote###DynamicField

Description:	Dynamic fields shown in the ticket note screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'DynamicField'} = {};`

6.43.28. Ticket::Frontend::AgentTicketNote###RichTextWidth

Description:	Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'RichTextWidth'} = '620';`

6.43.29. Ticket::Frontend::AgentTicketNote###RichTextHeight

Description:	Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'RichTextHeight'} = '100';`

6.44. Frontend::Agent::Ticket::ViewOwner

6.44.1. Ticket::Frontend::AgentTicketOwner###Permission

Description:	Required permissions to use the ticket owner screen of a zoomed ticket in the agent interface.

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Permission'} = 'owner';`

6.44.2. Ticket::Frontend::AgentTicketOwner###RequiredLock

Description:	Defines if a ticket lock is required in the ticket owner screen of a zoomed ticket in the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'RequiredLock'} = '0';`

6.44.3. Ticket::Frontend::AgentTicketOwner###TicketType

Description:	Sets the ticket type in the ticket owner screen of a zoomed ticket in the agent interface (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'TicketType'} = '0';`

6.44.4. Ticket::Frontend::AgentTicketOwner###Service

Description:	Sets the service in the ticket owner screen of a zoomed ticket in the agent interface (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Service'} =  '0';`

6.44.5. Ticket::Frontend::AgentTicketOwner###ServiceMandatory

Description:	Sets if service must be selected by the agent.

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'ServiceMandatory'} = '0';`

6.44.6. Ticket::Frontend::AgentTicketOwner###SLAMandatory

Description:	Sets if SLA must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'SLAMandatory'} = '0';`

6.44.7. Ticket::Frontend::AgentTicketOwner###Queue

Description:	Sets the queue in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Queue'} = '0';`

6.44.8. Ticket::Frontend::AgentTicketOwner###Owner

Description:	Sets the ticket owner in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Owner'} = '1';`

6.44.9. Ticket::Frontend::AgentTicketOwner###OwnerMandatory

Description:	Sets if ticket owner must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'OwnerMandatory'} = '1';`

6.44.10. Ticket::Frontend::AgentTicketOwner###Responsible

Description:	Sets the responsible agent of the ticket in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Responsible'} = '0';`

6.44.11. Ticket::Frontend::AgentTicketOwner###State

Description:	If a note is added by an agent, sets the state of the ticket in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'State'} = '0';`

6.44.12. Ticket::Frontend::AgentTicketOwner###StateType

Description:	Defines the next state of a ticket after adding a note, in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'StateType'} = [ 'open', 'pending reminder', 'pending auto' ];`

6.44.13. Ticket::Frontend::AgentTicketOwner###StateDefault

Description:	Defines the default next state of a ticket after adding a note, in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1

Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'StateDefault'} = 'open';`

6.44.14. Ticket::Frontend::AgentTicketOwner###Note

Description:	Allows adding notes in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Note'} = '1';`

6.44.15. Ticket::Frontend::AgentTicketOwner###NoteMandatory

Description:	Sets if note must be filled in by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'NoteMandatory'} = '1';`

6.44.16. Ticket::Frontend::AgentTicketOwner###Subject

Description:	Sets the default subject for notes added in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Subject'} = '$Text{"Owner Update"}!';`

6.44.17. Ticket::Frontend::AgentTicketOwner###Body

Description:	Sets the default body text for notes added in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Body'} = '';`

6.44.18. Ticket::Frontend::AgentTicketOwner###InvolvedAgent

Description:	Shows a list of all the involved agents on this ticket, in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'InvolvedAgent'} = '0';`

6.44.19. Ticket::Frontend::AgentTicketOwner###InformAgent

Description:	Shows a list of all the possible agents (all agents with note permissions on the queue/ticket) to determine who should be informed about this note, in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'InformAgent'} = '0';`

6.44.20. Ticket::Frontend::AgentTicketOwner###ArticleTypeDefault

Description:	Defines the default type of the note in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'ArticleTypeDefault'} = 'note-internal';`

6.44.21. Ticket::Frontend::AgentTicketOwner###ArticleTypes

Description:	Specifies the available note types for this ticket mask. If the option is deselected, ArticleTypeDefault is used and the option is removed from the mask.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'ArticleTypes'} = { 'note-external' => '0', 'note-internal' => '1', 'note-report' => '0' };`

6.44.22. Ticket::Frontend::AgentTicketOwner###Priority

Description:	Shows the ticket priority options in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Priority'} = '0';`

6.44.23. Ticket::Frontend::AgentTicketOwner###PriorityDefault

Description:	Defines the default ticket priority in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'PriorityDefault'} = '3 normal';`

6.44.24. Ticket::Frontend::AgentTicketOwner###Title

Description:	Shows the title fields in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Title'} = '0';`

6.44.25. Ticket::Frontend::AgentTicketOwner###HistoryType

Description:	Defines the history type for the ticket owner screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'HistoryType'} = 'AddNote';`

6.44.26. Ticket::Frontend::AgentTicketOwner###HistoryComment

Description:	Defines the history comment for the ticket owner screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'HistoryComment'} = '%%Owner';`

6.44.27. Ticket::Frontend::AgentTicketOwner###DynamicField

Description:	Dynamic fields shown in the ticket owner screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'DynamicField'} = {};`

6.44.28. Ticket::Frontend::AgentTicketOwner###RichTextWidth

Description:	Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'RichTextWidth'} = '620';`

6.44.29. Ticket::Frontend::AgentTicketOwner###RichTextHeight

Description:	Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1

Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'RichTextHeight'} = '100';`

6.45. Frontend::Agent::Ticket::ViewPending

6.45.1. Ticket::Frontend::AgentTicketPending###Permission

Description:	Required permissions to use the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Permission'} = 'pending';`

6.45.2. Ticket::Frontend::AgentTicketPending###RequiredLock

Description:	Defines if a ticket lock is required in the ticket pending screen of a zoomed ticket in the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'RequiredLock'} = '1';`

6.45.3. Ticket::Frontend::AgentTicketPending###TicketType

Description:	Sets the ticket type in the ticket pending screen of a zoomed ticket in the agent interface (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'TicketType'} = '0';`

6.45.4. Ticket::Frontend::AgentTicketPending###Service

Description:	Sets the service in the ticket pending screen of a zoomed ticket in the agent interface (Ticket::Service needs to be activated).

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Service'} = '0';`

6.45.5. Ticket::Frontend::AgentTicketPending###ServiceMandatory

Description:	Sets if service must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'ServiceMandatory'} = '0';`

6.45.6. Ticket::Frontend::AgentTicketPending###SLAMandatory

Description:	Sets if SLA must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'SLAMandatory'} = '0';`

6.45.7. Ticket::Frontend::AgentTicketPending###Queue

Description:	Sets the queue in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Queue'} = '0';`

6.45.8. Ticket::Frontend::AgentTicketPending###Owner

Description:	Sets the ticket owner in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending

Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Owner'} = '0';`

6.45.9. Ticket::Frontend::AgentTicketPending###OwnerMandatory

Description:	Sets if ticket owner must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'OwnerMandatory'} = '0';`

6.45.10. Ticket::Frontend::AgentTicketPending###Responsible

Description:	Sets the responsible agent of the ticket in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Responsible'} = '0';`

6.45.11. Ticket::Frontend::AgentTicketPending###State

Description:	If a note is added by an agent, sets the state of the ticket in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'State'} = '1';`

6.45.12. Ticket::Frontend::AgentTicketPending###StateType

Description:	Defines the next state of a ticket after adding a note, in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'StateType'} = [ 'pending reminder', 'pending auto' ];`

6.45.13. Ticket::Frontend::AgentTicketPending###StateDefault

Description:	Defines the default next state of a ticket after adding a note, in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'StateDefault'} = 'pending reminder';`

6.45.14. Ticket::Frontend::AgentTicketPending###Note

Description:	Allows adding notes in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Note'} = '1';`

6.45.15. Ticket::Frontend::AgentTicketPending###NoteMandatory

Description:	Sets if note must be filled in by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'NoteMandatory'} = '1';`

6.45.16. Ticket::Frontend::AgentTicketPending###Subject

Description:	Sets the default subject for notes added in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Subject'} = '$Text{"Pending"}!';`

6.45.17. Ticket::Frontend::AgentTicketPending###Body

Description:	Sets the default body text for notes added in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Body'} = '';`

6.45.18. Ticket::Frontend::AgentTicketPending###InvolvedAgent

Description:	Shows a list of all the involved agents on this ticket, in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'InvolvedAgent'} = '0';`

6.45.19. Ticket::Frontend::AgentTicketPending###InformAgent

Description:	Shows a list of all the possible agents (all agents with note permissions on the queue/ticket) to determine who should be informed about this note, in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'InformAgent'} = '0';`

6.45.20. Ticket::Frontend::AgentTicketPending###ArticleTypeDefault

Description:	Defines the default type of the note in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1

Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'ArticleTypeDefault'} = 'note-internal';`

6.45.21. Ticket::Frontend::AgentTicketPending###ArticleTypes

Description:	Specifies the available note types for this ticket mask. If the option is deselected, ArticleTypeDefault is used and the option is removed from the mask.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'ArticleTypes'} = { 'note-external' => '0', 'note-internal' => '1', 'note-report' => '0' };`

6.45.22. Ticket::Frontend::AgentTicketPending###Priority

Description:	Shows the ticket priority options in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Priority'} = '0';`

6.45.23. Ticket::Frontend::AgentTicketPending###PriorityDefault

Description:	Defines the default ticket priority in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'PriorityDefault'} = '3 normal';`

6.45.24. Ticket::Frontend::AgentTicketPending###Title

Description:	Shows the title fields in the ticket pending screen of a zoomed ticket in the agent interface.

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Title'} = '0';`

6.45.25. Ticket::Frontend::AgentTicketPending###HistoryType

Description:	Defines the history type for the ticket pending screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'HistoryType'} = 'AddNote';`

6.45.26. Ticket::Frontend::AgentTicketPending###HistoryComment

Description:	Defines the history comment for the ticket pending screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'HistoryComment'} = '%%Pending';`

6.45.27. Ticket::Frontend::AgentTicketPending###DynamicField

Description:	Dynamic fields shown in the ticket pending screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'DynamicField'} = {};`

6.45.28. Ticket::Frontend::AgentTicketPending###RichTextWidth

Description:	Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'RichTextWidth'} = '620';`

6.45.29. Ticket::Frontend::AgentTicketPending###RichTextHeight

Description:	Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'RichTextHeight'} = '100';`

6.46. Frontend::Agent::Ticket::ViewPhoneInbound

6.46.1. Ticket::Frontend::AgentTicketPhoneInbound###Permission

Description:	Required permissions to use the ticket phone inbound screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'Permission'} = 'phone';`

6.46.2. Ticket::Frontend::AgentTicketPhoneInbound###RequiredLock

Description:	Defines if a ticket lock is required in the ticket phone inbound screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'RequiredLock'} = '0';`

6.46.3. Ticket::Frontend::AgentTicketPhoneInbound###ArticleType

Description:	Defines the default type of the note in the ticket phone inbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'ArticleType'} = 'phone';`

6.46.4. Ticket::Frontend::AgentTicketPhoneInbound###SenderType

Description:	Defines the default sender type for phone tickets in the ticket phone inbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'SenderType'} = 'customer';`

6.46.5. Ticket::Frontend::AgentTicketPhoneInbound###Subject

Description:	Defines the default subject for phone tickets in the ticket phone inbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'Subject'} = '$Text{"Phone call"}!';`

6.46.6. Ticket::Frontend::AgentTicketPhoneInbound###Body

Description:	Defines the default note body text for phone tickets in the ticket phone inbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'Body'} = '';`

6.46.7. Ticket::Frontend::AgentTicketPhoneInbound###State

Description:	Defines the default ticket next state after adding a phone note in the ticket phone inbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'State'} = 'open';`

6.46.8. Ticket::Frontend::AgentTicketPhoneInbound###StateType

Description:	Next possible ticket states after adding a phone note in the ticket phone inbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'StateType'} = [ 'open', 'pending auto', 'pending reminder', 'closed' ];`

6.46.9. Ticket::Frontend::AgentTicketPhoneInbound###HistoryType

Description:	Defines the history type for the ticket phone inbound screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'HistoryType'} = 'PhoneCallCustomer';`

6.46.10. Ticket::Frontend::AgentTicketPhoneInbound###HistoryCommen

Description:	Defines the history comment for the ticket phone inbound screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'HistoryComment'} = '';`

6.46.11. Ticket::Frontend::AgentTicketPhoneInbound###DynamicField

Description:	Dynamic fields shown in the ticket phone inbound screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'DynamicField'} = {};`

6.46.12. Ticket::Frontend::AgentTicketPhoneInbound###RichTextWidth

Description:	Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'RichTextWidth'} = '475';`

6.46.13. Ticket::Frontend::AgentTicketPhoneInbound###RichTextHeight

Description:	Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'RichTextHeight'} = '200';`

6.47. Frontend::Agent::Ticket::ViewPhoneNew

6.47.1. Ticket::Frontend::AgentTicketPhone###Priority

Description:	Sets the default priority for new phone tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew

Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'Priority'} = '3 normal';`

6.47.2. Ticket::Frontend::AgentTicketPhone###ArticleType

Description:	Sets the default article type for new phone tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'ArticleType'} = 'phone';`

6.47.3. Ticket::Frontend::AgentTicketPhone###SenderType

Description:	Sets the default sender type for new phone ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'SenderType'} = 'customer';`

6.47.4. Ticket::Frontend::AgentTicketPhone::AllowMultipleFrom

Description:	Controls if more than one from entry can be set in the new phone ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone::AllowMultipleFrom'} = '1';`

6.47.5. Ticket::Frontend::AgentTicketPhone###Subject

Description:	Sets the default subject for new phone tickets (e.g. 'Phone call') in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1

Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'Subject'} = '';`

6.47.6. Ticket::Frontend::AgentTicketPhone###Body

Description:	Sets the default note text for new telephone tickets. E.g 'New ticket via call' in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'Body'} = '';`

6.47.7. Ticket::Frontend::AgentTicketPhone###StateDefault

Description:	Sets the default next state for new phone tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'StateDefault'} = 'open';`

6.47.8. Ticket::Frontend::AgentTicketPhone###StateType

Description:	Determines the next possible ticket states, after the creation of a new phone ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'StateType'} = [ 'open', 'pending auto', 'pending reminder', 'closed' ];`

6.47.9. Ticket::Frontend::AgentTicketPhone###HistoryType

Description:	Defines the history type for the phone ticket screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'HistoryType'} = 'PhoneCallCustomer';`

6.47.10. Ticket::Frontend::AgentTicketPhone###HistoryComment

Description:	Defines the history comment for the phone ticket screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'HistoryComment'} = '';`

6.47.11. Ticket::Frontend::AgentTicketPhone###SplitLinkType

Description:	Sets the default link type of splitted tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'SplitLinkType'} = { 'Direction' => 'Target', 'LinkType' => 'ParentChild' };`

6.47.12. Ticket::Frontend::AgentTicketPhone###ServiceMandatory

Description:	Sets if service must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'ServiceMandatory'} = '0';`

6.47.13. Ticket::Frontend::AgentTicketPhone###SLAMandatory

Description:	Sets if SLA must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'SLAMandatory'} = '0';`

6.47.14. Ticket::Frontend::AgentTicketPhone###DynamicField

Description:	Dynamic fields shown in the ticket phone screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'DynamicField'} = {};`

6.47.15. Ticket::Frontend::AgentTicketPhone###RichTextWidth

Description:	Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'RichTextWidth'} = '620';`

6.47.16. Ticket::Frontend::AgentTicketPhone###RichTextHeight

Description:	Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'RichTextHeight'} = '320';`

6.48. Frontend::Agent::Ticket::ViewPhoneOutbound

6.48.1. Ticket::Frontend::AgentTicketPhoneOutbound###Permission

Description:	Required permissions to use the ticket phone outbound screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1

Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'Permission'} = 'phone';`

6.48.2. Ticket::Frontend::AgentTicketPhoneOutbound###RequiredLock

Description:	Defines if a ticket lock is required in the ticket phone outbound screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'RequiredLock'} = '1';`

6.48.3. Ticket::Frontend::AgentTicketPhoneOutbound###ArticleType

Description:	Defines the default type of the note in the ticket phone outbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'ArticleType'} = 'phone';`

6.48.4. Ticket::Frontend::AgentTicketPhoneOutbound###SenderType

Description:	Defines the default sender type for phone tickets in the ticket phone outbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'SenderType'} = 'agent';`

6.48.5. Ticket::Frontend::AgentTicketPhoneOutbound###Subject

Description:	Defines the default subject for phone tickets in the ticket phone outbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound

Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'Subject'} = '$Text{"Phone call"}!';`

6.48.6. Ticket::Frontend::AgentTicketPhoneOutbound###Body

Description:	Defines the default note body text for phone tickets in the ticket phone outbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'Body'} = '';`

6.48.7. Ticket::Frontend::AgentTicketPhoneOutbound###State

Description:	Defines the default ticket next state after adding a phone note in the ticket phone outbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'State'} = 'closed successful';`

6.48.8. Ticket::Frontend::AgentTicketPhoneOutbound###StateType

Description:	Next possible ticket states after adding a phone note in the ticket phone outbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'StateType'} = [ 'open', 'pending auto', 'pending reminder', 'closed' ];`

6.48.9. Ticket::Frontend::AgentTicketPhoneOutbound###HistoryType

Description:	Defines the history type for the ticket phone outbound screen action, which gets used for ticket history in the agent interface.

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'HistoryType'} = 'PhoneCallAgent';`

6.48.10. Ticket::Frontend::AgentTicketPhoneOutbound###HistoryComme

Description:	Defines the history comment for the ticket phone outbound screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'HistoryComment'} = '';`

6.48.11. Ticket::Frontend::AgentTicketPhoneOutbound###DynamicField

Description:	Dynamic fields shown in the ticket phone outbound screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'DynamicField'} = {};`

6.48.12. Ticket::Frontend::AgentTicketPhoneOutbound###RichTextWidth

Description:	Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'RichTextWidth'} = '475';`

6.48.13. Ticket::Frontend::AgentTicketPhoneOutbound###RichTextHeigh

Description:	Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'RichTextHeight'} = '200';`

6.49. Frontend::Agent::Ticket::ViewPrint

6.49.1. Ticket::Frontend::AgentTicketPrint###DynamicField

Description:	Dynamic fields shown in the ticket print screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPrint
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPrint'}->{'DynamicField'} = {};`

6.50. Frontend::Agent::Ticket::ViewPriority

6.50.1. Ticket::Frontend::AgentTicketPriority###Permission

Description:	Required permissions to use the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Permission'} = 'priority';`

6.50.2. Ticket::Frontend::AgentTicketPriority###RequiredLock

Description:	Defines if a ticket lock is required in the ticket priority screen of a zoomed ticket in the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'RequiredLock'} = '1';`

6.50.3. Ticket::Frontend::AgentTicketPriority###TicketType

Description:	Sets the ticket type in the ticket priority screen of a zoomed ticket in the agent interface (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'TicketType'} = '0';`

6.50.4. Ticket::Frontend::AgentTicketPriority###Service

Description:	Sets the service in the ticket priority screen of a zoomed ticket in the agent interface (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Service'} = '0';`

6.50.5. Ticket::Frontend::AgentTicketPriority###ServiceMandatory

Description:	Sets if service must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'ServiceMandatory'} = '0';`

6.50.6. Ticket::Frontend::AgentTicketPriority###SLAMandatory

Description:	Sets if SLA must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'SLAMandatory'} = '0';`

6.50.7. Ticket::Frontend::AgentTicketPriority###Queue

Description:	Sets the queue in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Queue'} = '0';`

6.50.8. Ticket::Frontend::AgentTicketPriority###Owner

Description:	Sets the ticket owner in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Owner'} = '0';`

6.50.9. Ticket::Frontend::AgentTicketPriority###OwnerMandatory

Description:	Sets if ticket owner must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'OwnerMandatory'} = '0';`

6.50.10. Ticket::Frontend::AgentTicketPriority###Responsible

Description:	Sets the responsible agent of the ticket in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Responsible'} = '0';`

6.50.11. Ticket::Frontend::AgentTicketPriority###State

Description:	If a note is added by an agent, sets the state of the ticket in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'State'} = '0';`

6.50.12. Ticket::Frontend::AgentTicketPriority###StateType

Description:	Defines the next state of a ticket after adding a note, in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'StateType'} = [ 'open', 'pending reminder', 'pending auto' ];`

6.50.13. Ticket::Frontend::AgentTicketPriority###StateDefault

Description:	Defines the default next state of a ticket after adding a note, in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'StateDefault'} = 'open';`

6.50.14. Ticket::Frontend::AgentTicketPriority###Note

Description:	Allows adding notes in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Note'} = '1';`

6.50.15. Ticket::Frontend::AgentTicketPriority###NoteMandatory

Description:	Sets if note must be filled in by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'NoteMandatory'} = '1';`

6.50.16. Ticket::Frontend::AgentTicketPriority###Subject

Description:	Sets the default subject for notes added in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Subject'} = '$Text{"Priority Update"}!';`

6.50.17. Ticket::Frontend::AgentTicketPriority###Body

Description:	Sets the default body text for notes added in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Body'} = '';`

6.50.18. Ticket::Frontend::AgentTicketPriority###InvolvedAgent

Description:	Shows a list of all the involved agents on this ticket, in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'InvolvedAgent'} = '0';`

6.50.19. Ticket::Frontend::AgentTicketPriority###InformAgent

Description:	Shows a list of all the possible agents (all agents with note permissions on the queue/ticket) to determine who should be informed about this note, in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'InformAgent'} = '0';`

6.50.20. Ticket::Frontend::AgentTicketPriority###ArticleTypeDefault

Description:	Defines the default type of the note in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'ArticleTypeDefault'} = 'note-internal';`

6.50.21. Ticket::Frontend::AgentTicketPriority###ArticleTypes

Description:	Specifies the available note types for this ticket mask. If the option is deselected, ArticleTypeDefault is used and the option is removed from the mask.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'ArticleTypes'} = { 'note-external' => '0', 'note-internal' => '1', 'note-report' => '0' };`

6.50.22. Ticket::Frontend::AgentTicketPriority###Priority

Description:	Shows the ticket priority options in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority

Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Priority'} = '1';`

6.50.23. Ticket::Frontend::AgentTicketPriority###PriorityDefault

Description:	Defines the default ticket priority in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'PriorityDefault'} = '3 normal';`

6.50.24. Ticket::Frontend::AgentTicketPriority###Title

Description:	Shows the title fields in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Title'} = '0';`

6.50.25. Ticket::Frontend::AgentTicketPriority###HistoryType

Description:	Defines the history type for the ticket priority screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'HistoryType'} = 'AddNote';`

6.50.26. Ticket::Frontend::AgentTicketPriority###HistoryComment

Description:	Defines the history comment for the ticket priority screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1

Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'HistoryComment'} = '%%Priority';`

6.50.27. Ticket::Frontend::AgentTicketPriority###DynamicField

Description:	Dynamic fields shown in the ticket priority screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'DynamicField'} = {};`

6.50.28. Ticket::Frontend::AgentTicketPriority###RichTextWidth

Description:	Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'RichTextWidth'} = '620';`

6.50.29. Ticket::Frontend::AgentTicketPriority###RichTextHeight

Description:	Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'RichTextHeight'} = '100';`

6.51. Frontend::Agent::Ticket::ViewQueue

6.51.1. Ticket::Frontend::AgentTicketQueue###StripEmptyLines

Description:	Strips empty lines on the ticket preview in the queue view.
Group:	Ticket

SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'StripEmptyLines'} = '0';`

6.51.2. Ticket::Frontend::AgentTicketQueue###ViewAllPossibleTickets

Description:	Shows all both ro and rw queues in the queue view.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'ViewAllPossibleTickets'} = '0';`

6.51.3. Ticket::Frontend::AgentTicketQueue###HighlightAge1

Description:	Sets the age in minutes (first level) for highlighting queues that contain untouched tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'HighlightAge1'} = '1440';`

6.51.4. Ticket::Frontend::AgentTicketQueue###HighlightAge2

Description:	Sets the age in minutes (second level) for highlighting queues that contain untouched tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'HighlightAge2'} = '2880';`

6.51.5. Ticket::Frontend::AgentTicketQueue###Blink

Description:	Activates a blinking mechanism of the queue that contains the oldest ticket.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1

Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'Blink'} = '1';`

6.51.6. Ticket::Frontend::AgentTicketQueue###QueueSort

Description:	Sorts the tickets (ascendingly or descendingly) when a single queue is selected in the queue view and after the tickets are sorted by priority. Values: 0 = ascending (oldest on top, default), 1 = descending (youngest on top). Use the QueueID for the key and 0 or 1 for value.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'QueueSort'} = { '3' => '0', '7' => '1' };`

6.51.7. Ticket::Frontend::AgentTicketQueue###SortBy::Default

Description:	Defines the default sort criteria for all queues displayed in the queue view.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'SortBy::Default'} = 'Age';`

6.51.8. Ticket::Frontend::AgentTicketQueue###PreSort::ByPriority

Description:	Defines if a pre-sorting by priority should be done in the queue view.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'PreSort::ByPriority'} = '1';`

6.51.9. Ticket::Frontend::AgentTicketQueue###Order::Default

Description:	Defines the default sort order for all queues in the queue view, after priority sort.

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'Order::Default'} = 'Up';`

6.51.10. Ticket::Frontend::AgentTicketQueue###DefaultColumns

Description:	Columns that can be filtered in the queue view of the agent interface. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'DefaultColumns'} = { 'Age' => '2', 'Changed' => '1', 'CustomerID' => '2', 'CustomerName' => '1', 'CustomerUserID' => '1', 'EscalationResponseTime' => '1', 'EscalationSolutionTime' => '1', 'EscalationTime' => '1', 'EscalationUpdateTime' => '1', 'Lock' => '2', 'Owner' => '2', 'PendingTime' => '1', 'Priority' => '1', 'Queue' => '2', 'Responsible' => '1', 'SLA' => '1', 'Service' => '1', 'State' => '2', 'TicketNumber' => '2', 'Title' => '2', 'Type' => '1' };`

6.52. Frontend::Agent::Ticket::ViewResponsible

6.52.1. Ticket::Frontend::AgentTicketResponsibleView###SortBy::Defau

Description:	Defines the default ticket attribute for ticket sorting in the responsible view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsibleView'}->{'SortBy::Default'} = 'Age';`

6.52.2. Ticket::Frontend::AgentTicketResponsibleView###Order::Defau

Description:	Defines the default ticket order in the responsible view of the agent interface. Up: oldest on top. Down: latest on top.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsibleView'}->{'Order::Default'} = 'Up';`

6.52.3. Ticket::Frontend::AgentTicketResponsible###Permission

Description:	Required permissions to use the ticket responsible screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Permission'} = 'responsible';`

6.52.4. Ticket::Frontend::AgentTicketResponsible###RequiredLock

Description:	Defines if a ticket lock is required in the ticket responsible screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'RequiredLock'} = '0';`

6.52.5. Ticket::Frontend::AgentTicketResponsible###TicketType

Description:	Sets the ticket type in the ticket responsible screen of the agent interface (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1

Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'TicketType'} = '0';`

6.52.6. Ticket::Frontend::AgentTicketResponsible###Service

Description:	Sets the service in the ticket responsible screen of the agent interface (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Service'} = '0';`

6.52.7. Ticket::Frontend::AgentTicketResponsible###ServiceMandatory

Description:	Sets if service must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'ServiceMandatory'} = '0';`

6.52.8. Ticket::Frontend::AgentTicketResponsible###SLAMandatory

Description:	Sets if SLA must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'SLAMandatory'} = '0';`

6.52.9. Ticket::Frontend::AgentTicketResponsible###Queue

Description:	Sets the queue in the ticket responsible screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Queue'} = '0';`

6.52.10. Ticket::Frontend::AgentTicketResponsible###Owner

Description:	Sets the ticket owner in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Owner'} = '0';`

6.52.11. Ticket::Frontend::AgentTicketResponsible###OwnerMandatory

Description:	Sets if ticket owner must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'OwnerMandatory'} = '0';`

6.52.12. Ticket::Frontend::AgentTicketResponsible###Responsible

Description:	Sets the responsible agent of the ticket in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Responsible'} = '1';`

6.52.13. Ticket::Frontend::AgentTicketResponsible###State

Description:	If a note is added by an agent, sets the state of a ticket in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'State'} = '0';`

6.52.14. Ticket::Frontend::AgentTicketResponsible###StateType

Description:	Defines the next state of a ticket after adding a note, in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'StateType'} = [ 'open', 'pending reminder', 'pending auto' ];`

6.52.15. Ticket::Frontend::AgentTicketResponsible###StateDefault

Description:	Defines the default next state of a ticket after adding a note, in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'StateDefault'} = 'open';`

6.52.16. Ticket::Frontend::AgentTicketResponsible###Note

Description:	Allows adding notes in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Note'} = '1';`

6.52.17. Ticket::Frontend::AgentTicketResponsible###NoteMandatory

Description:	Sets if note must be filled in by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'NoteMandatory'} = '1';`

6.52.18. Ticket::Frontend::AgentTicketResponsible###Subject

Description:	Sets the default subject for notes added in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Subject'} = '$Text{"Responsible Update"}!';`

6.52.19. Ticket::Frontend::AgentTicketResponsible###Body

Description:	Sets the default body text for notes added in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Body'} = '';`

6.52.20. Ticket::Frontend::AgentTicketResponsible###InvolvedAgent

Description:	Shows a list of all the involved agents on this ticket, in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'InvolvedAgent'} = '0';`

6.52.21. Ticket::Frontend::AgentTicketResponsible###InformAgent

Description:	Shows a list of all the possible agents (all agents with note permissions on the queue/ticket) to determine who should be informed about this note, in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	0

Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'InformAgent'} = '0';`

6.52.22. Ticket::Frontend::AgentTicketResponsible###ArticleTypeDefault

Description:	Defines the default type of the note in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'ArticleTypeDefault'} = 'note-internal';`

6.52.23. Ticket::Frontend::AgentTicketResponsible###ArticleTypes

Description:	Specifies the available note types for this ticket mask. If the option is deselected, ArticleTypeDefault is used and the option is removed from the mask.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'ArticleTypes'} = { 'note-external' => '0', 'note-internal' => '1', 'note-report' => '0' };`

6.52.24. Ticket::Frontend::AgentTicketResponsible###Priority

Description:	Shows the ticket priority options in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Priority'} = '0';`

6.52.25. Ticket::Frontend::AgentTicketResponsible###PriorityDefault

Description:	Defines the default ticket priority in the ticket responsible screen of the agent interface.
Group:	Ticket

SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'PriorityDefault'} = '3 normal';`

6.52.26. Ticket::Frontend::AgentTicketResponsible###Title

Description:	Shows the title fields in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Title'} = '1';`

6.52.27. Ticket::Frontend::AgentTicketResponsible###HistoryType

Description:	Defines the history type for the ticket responsible screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'HistoryType'} = 'AddNote';`

6.52.28. Ticket::Frontend::AgentTicketResponsible###HistoryComment

Description:	Defines the history comment for the ticket responsible screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'HistoryComment'} = '%%Responsible';`

6.52.29. Ticket::Frontend::AgentTicketResponsible###DynamicField

Description:	Dynamic fields shown in the ticket responsible screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket

SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'DynamicField'} = {};`

6.52.30. Ticket::Frontend::AgentTicketResponsible###RichTextWidth

Description:	Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'RichTextWidth'} = '620';`

6.52.31. Ticket::Frontend::AgentTicketResponsible###RichTextHeight

Description:	Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'RichTextHeight'} = '100';`

6.52.32. Ticket::Frontend::AgentTicketResponsibleView###DefaultColum

Description:	Columns that can be filtered in the responsible view of the agent interface. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsibleView'}->{'DefaultColumns'} = { 'Age' => '2', 'Changed' => '1', 'CustomerID' => '2', 'CustomerName' => '1', 'CustomerUserID' => '1', 'EscalationResponseTime' => '1', 'EscalationSolutionTime' => '1', 'EscalationTime' => '1',`

```
  'EscalationUpdateTime' => '1',
  'Lock' => '2',
  'Owner' => '2',
  'PendingTime' => '1',
  'Priority' => '1',
  'Queue' => '2',
  'Responsible' => '1',
  'SLA' => '1',
  'Service' => '1',
  'State' => '2',
  'TicketNumber' => '2',
  'Title' => '2',
  'Type' => '1'
};
```

6.53. Frontend::Agent::Ticket::ViewSearch

6.53.1. Ticket::Frontend::AgentTicketSearch###ExtendedSearchConditi

Description:	Allows extended search conditions in ticket search of the agent interface. With this feature you can search e. g. with this kind of conditions like "(key1&&key2)" or "(key1||key2)".
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'ExtendedSearchCondition'} = '1';`

6.53.2. Ticket::Frontend::AgentTicketSearch###SearchLimit

Description:	Maximum number of tickets to be displayed in the result of a search in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchLimit'} = '2000';`

6.53.3. Ticket::Frontend::AgentTicketSearch###SearchPageShown

Description:	Number of tickets to be displayed in each page of a search result in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchPageShown'} = '40';`

6.53.4. Ticket::Frontend::AgentTicketSearch###SearchViewableTicketL

Description:	Number of lines (per ticket) that are shown by the search utility in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchViewableTicketLines'} = '10';`

6.53.5. Ticket::Frontend::AgentTicketSearch###SortBy::Default

Description:	Defines the default ticket attribute for ticket sorting of the ticket search result of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SortBy::Default'} = 'Age';`

6.53.6. Ticket::Frontend::AgentTicketSearch###Order::Default

Description:	Defines the default ticket order in the ticket search result of the agent interface. Up: oldest on top. Down: latest on top.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Order::Default'} = 'Down';`

6.53.7. Ticket::Frontend::AgentTicketSearch###SearchArticleCSVTree

Description:	Exports the whole article tree in search result (it can affect the system performance).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchArticleCSVTree'} = '0';`

6.53.8. Ticket::Frontend::AgentTicketSearch###SearchCSVData

Description:	Data used to export the search result in CSV format.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	<pre>$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'];</pre>

6.53.9. Ticket::Frontend::AgentTicketSearch###ArticleCreateTime

Description:	Includes article create times in the ticket search of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	0
Config-Setting:	<pre>$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'ArticleCreateTime'} = '0';</pre>

6.53.10. Ticket::Frontend::AgentTicketSearch###Defaults###Fulltext

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'Fulltext'} = '';`

6.53.11. Ticket::Frontend::AgentTicketSearch###Defaults###TicketNum

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketNumber'} = '';`

6.53.12. Ticket::Frontend::AgentTicketSearch###Defaults###Title

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'Title'} = '';`

6.53.13. Ticket::Frontend::AgentTicketSearch###Defaults###From

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'From'} = '';`

6.53.14. Ticket::Frontend::AgentTicketSearch###Defaults###To

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'To'} = '';`

6.53.15. Ticket::Frontend::AgentTicketSearch###Defaults###Cc

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'Cc'} = '';`

6.53.16. Ticket::Frontend::AgentTicketSearch###Defaults###Subject

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'Subject'} = '';`

6.53.17. Ticket::Frontend::AgentTicketSearch###Defaults###Body

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'Body'} = '';`

6.53.18. Ticket::Frontend::AgentTicketSearch###Defaults###CustomerI

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'CustomerID'} = '';`

6.53.19. Ticket::Frontend::AgentTicketSearch###Defaults###CustomerU

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'CustomerUserLogin'} = '';`

6.53.20. Ticket::Frontend::AgentTicketSearch###Defaults###StateIDs

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'StateIDs'} = [];`

6.53.21. Ticket::Frontend::AgentTicketSearch###Defaults###QueueIDs

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'QueueIDs'} = [];`

6.53.22. Ticket::Frontend::AgentTicketSearch###Defaults###TicketCrea

Description:	Default data to use on attribute for ticket search screen. Example: "TicketCreateTimePointFormat=year;TicketCreateTimePointStart=Last;TicketCre
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketCreateTimePoint'} = '';`

6.53.23. Ticket::Frontend::AgentTicketSearch###Defaults###TicketCrea

Description:	Default data to use on attribute for ticket search screen. Example: "TicketCreateTimeStartYear=2010;TicketCreateTimeStartMonth=10;TicketCreate
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketCreateTimeSlot'} = '';`

6.53.24. Ticket::Frontend::AgentTicketSearch###Defaults###TicketChar

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketChangeTimePoint'} = '';`

6.53.25. Ticket::Frontend::AgentTicketSearch###Defaults###TicketChar

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketChangeTimeSlot'} = '';`

6.53.26. Ticket::Frontend::AgentTicketSearch###Defaults###TicketClos

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketCloseTimePoint'} = '';`

6.53.27. Ticket::Frontend::AgentTicketSearch###Defaults###TicketClos

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketCloseTimeSlot'} = '';`

6.53.28. Ticket::Frontend::AgentTicketSearch###Defaults###TicketEsca

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketEscalationTimePoint'} = '';`

6.53.29. Ticket::Frontend::AgentTicketSearch###Defaults###TicketEsca

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketEscalationTimeSlot'} = '';`

6.53.30. Ticket::Frontend::AgentTicketSearch###Defaults###ArticleCrea

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'ArticleCreateTimePoint'} = '';`

6.53.31. Ticket::Frontend::AgentTicketSearch###Defaults###ArticleCrea

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'ArticleCreateTimeSlot'} = '';`

6.53.32. Ticket::Frontend::AgentTicketSearch###Defaults###SearchInA

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'SearchInArchive'} = '';`

6.53.33. Ticket::Frontend::CustomerTicketSearch###SearchArticleCSVTr

Description:	Exports the whole article tree in search result (it can affect the system performance).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketSearch'}->{'SearchArticleCSVTree'} = '0';`

6.53.34. Ticket::Frontend::AgentTicketSearch###DynamicField

Description:	Dynamic fields shown in the ticket search screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and shown by default.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'DynamicField'} = {};`

6.53.35. Ticket::Frontend::AgentTicketSearch###Defaults###DynamicFi

Description:	Defines the default shown ticket search attribute for ticket search screen. Example: "Key" must have the name of the Dynamic Field in this case 'X', "Content" must have the value of the Dynamic Field depending on the Dynamic Field type, Text: 'a text', Dropdown: '1', Date/Time: 'Search_DynamicField_XTimeSlotStartYear=1974; Search_DynamicField_XTimeSlotStartMonth=01; Search_DynamicField_XTimeSlotStartDay=26; Search_DynamicField_XTimeSlotStartHour=00; Search_DynamicField_XTimeSlotStartMinute=00; Search_DynamicField_XTimeSlotStartSecond=00; Search_DynamicField_XTimeSlotStopYear=2013; Search_DynamicField_XTimeSlotStopMonth=01; Search_DynamicField_XTimeSlotStopDay=26; Search_DynamicField_XTimeSlotStopHour=23; Search_DynamicField_XTimeSlotStopMinute=59; Search_DynamicField_XTimeSlotStopSecond=59;' and or 'Search_DynamicField_XTimePointFormat=week; Search_DynamicField_XTimePointStart=Before; Search_DynamicField_XTimePointValue=7';.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'DynamicField'} = {};`

6.53.36. Ticket::Frontend::AgentTicketSearch###SearchCSVDynamicFiel

Description:	Dynamic Fields used to export the search result in CSV format.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchCSVDynamicField'} = {};`

6.53.37. Ticket::Frontend::AgentTicketSearch###DefaultColumns

Description:	Columns that can be filtered in the ticket search result view of the agent interface. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'DefaultColumns'} = {
	'Age' => '2',
	'Changed' => '1',
	'CustomerID' => '2',
	'CustomerName' => '1',
	'CustomerUserID' => '1',
	'EscalationResponseTime' => '1',
	'EscalationSolutionTime' => '1',
	'EscalationTime' => '1',
	'EscalationUpdateTime' => '1',
	'Lock' => '2',
	'Owner' => '2',
	'PendingTime' => '1',
	'Priority' => '1',
	'Queue' => '2',
	'Responsible' => '1',
	'SLA' => '1',
	'Service' => '1',
	'State' => '2',
	'TicketNumber' => '2',
	'Title' => '2',
	'Type' => '1'
	};</pre>

6.54. Frontend::Agent::Ticket::ViewStatus

6.54.1. Ticket::Frontend::AgentTicketStatusView###ViewableTicketsPa

Description:	Shows all open tickets (even if they are locked) in the status view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewStatus
Valid:	1
Required:	1
Config-Setting:	<pre>$Self->{'Ticket::Frontend::AgentTicketStatusView'}->{'ViewableTicketsPage'} = '50';</pre>

6.54.2. Ticket::Frontend::AgentTicketStatusView###SortBy::Default

Description:	Defines the default ticket attribute for ticket sorting in the status view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewStatus
Valid:	1
Required:	1
Config-Setting:	<pre>$Self->{'Ticket::Frontend::AgentTicketStatusView'}->{'SortBy::Default'} = 'Age';</pre>

6.54.3. Ticket::Frontend::AgentTicketStatusView###Order::Default

Description:	Defines the default ticket order (after priority sort) in the status view of the agent interface. Up: oldest on top. Down: latest on top.

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewStatus
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketStatusView'}->{'Order::Default'} = 'Down';`

6.54.4. Ticket::Frontend::AgentTicketStatusView###DefaultColumns

Description:	Columns that can be filtered in the status view of the agent interface. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewStatus
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketStatusView'}->{'DefaultColumns'} = { 'Age' => '2', 'Changed' => '1', 'CustomerID' => '2', 'CustomerName' => '1', 'CustomerUserID' => '1', 'EscalationResponseTime' => '1', 'EscalationSolutionTime' => '1', 'EscalationTime' => '1', 'EscalationUpdateTime' => '1', 'Lock' => '2', 'Owner' => '2', 'PendingTime' => '1', 'Priority' => '1', 'Queue' => '2', 'Responsible' => '1', 'SLA' => '1', 'Service' => '1', 'State' => '2', 'TicketNumber' => '2', 'Title' => '2', 'Type' => '1' };`

6.55. Frontend::Agent::Ticket::ViewWatch

6.55.1. Ticket::Frontend::AgentTicketWatchView###SortBy::Default

Description:	Defines the default ticket attribute for ticket sorting in the watch view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewWatch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketWatchView'}->{'SortBy::Default'} = 'Age';`

6.55.2. Ticket::Frontend::AgentTicketWatchView###Order::Default

Description:	Defines the default ticket order in the watch view of the agent interface. Up: oldest on top. Down: latest on top.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewWatch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketWatchView'}->{'Order::Default'} = 'Up';`

6.55.3. Ticket::Frontend::AgentTicketWatchView###DefaultColumns

Description:	Columns that can be filtered in the watch view of the agent interface. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewWatch
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketWatchView'}->{'DefaultColumns'} = { 'Age' => '2', 'Changed' => '1', 'CustomerID' => '2', 'CustomerName' => '1', 'CustomerUserID' => '1', 'EscalationResponseTime' => '1', 'EscalationSolutionTime' => '1', 'EscalationTime' => '1', 'EscalationUpdateTime' => '1', 'Lock' => '2', 'Owner' => '2', 'PendingTime' => '1', 'Priority' => '1', 'Queue' => '2', 'Responsible' => '1', 'SLA' => '1', 'Service' => '1', 'State' => '2', 'TicketNumber' => '2', 'Title' => '2', 'Type' => '1' };`

6.56. Frontend::Agent::Ticket::ViewZoom

6.56.1. Ticket::Frontend::PlainView

Description:	Shows a link to see a zoomed email ticket in plain text.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1

Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::PlainView'} = '0';`

6.56.2. Ticket::Frontend::ZoomExpand

Description:	Shows all the articles of the ticket (expanded) in the zoom view.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ZoomExpand'} = '0';`

6.56.3. Ticket::Frontend::ZoomExpandSort

Description:	Shows the articles sorted normally or in reverse, under ticket zoom in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ZoomExpandSort'} = 'normal';`

6.56.4. Ticket::ZoomAttachmentDisplayCount

Description:	Shows a count of icons in the ticket zoom, if the article has attachments.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::ZoomAttachmentDisplayCount'} = '20';`

6.56.5. Ticket::ZoomTimeDisplay

Description:	Displays the accounted time for an article in the ticket zoom view.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::ZoomTimeDisplay'} = '0';`

6.56.6. Ticket::UseArticleColors

Description:	Shows colors for different article types in the article table.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1

Required:	1
Config-Setting:	`$Self->{'Ticket::UseArticleColors'} = '0';`

6.56.7. Ticket::Frontend::TicketArticleFilter

Description:	Activates the article filter in the zoom view to specify which articles should be shown.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::TicketArticleFilter'} = '0';`

6.56.8. Ticket::Frontend::HTMLArticleHeightDefault

Description:	Set the default height (in pixels) of inline HTML articles in AgentTicketZoom.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::HTMLArticleHeightDefault'} = '100';`

6.56.9. Ticket::Frontend::HTMLArticleHeightMax

Description:	Set the maximum height (in pixels) of inline HTML articles in AgentTicketZoom.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::HTMLArticleHeightMax'} = '2500';`

6.56.10. Ticket::Frontend::ZoomRichTextForce

Description:	Show article as rich text even if rich text writing is disabled.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ZoomRichTextForce'} = '1';`

6.56.11. Ticket::Frontend::AgentTicketZoom###DynamicField

Description:	Dynamic fields shown in the sidebar of the ticket zoom screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled.

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketZoom'}->{'DynamicField'} = {};`

6.57. Frontend::Agent::TicketOverview

6.57.1. Ticket::Frontend::Overview###Small

Description:	Allows having a small format ticket overview (CustomerInfo => 1 - shows also the customer information).
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::Overview'}->{'Small'} = { 'CustomerInfo' => '1', 'Module' => 'Kernel::Output::HTML::TicketOverviewSmall', 'ModulePriority' => '100', 'Name' => 'Small', 'NameShort' => 'S' };`

6.57.2. Ticket::Frontend::OverviewSmall###ColumnHeader

Description:	Shows either the last customer article's subject or the ticket title in the small format overview.
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::OverviewSmall'}->{'ColumnHeader'} = 'LastCustomerSubject';`

6.57.3. Ticket::Frontend::Overview###Medium

Description:	Allows having a medium format ticket overview (CustomerInfo => 1 - shows also the customer information).
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::Overview'}->{'Medium'} = { 'CustomerInfo' => '0', 'Module' => 'Kernel::Output::HTML::TicketOverviewMedium', 'ModulePriority' => '200', 'Name' => 'Medium', 'NameShort' => 'M',`

```
  'OverviewMenuModules' => '1',
  'TicketActionsPerTicket' => '0'
};
```

6.57.4. Ticket::Frontend::Overview###Preview

Description:	Shows a preview of the ticket overview (CustomerInfo => 1 - shows also Customer-Info, CustomerInfoMaxSize max. size in characters of Customer-Info).
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'Ticket::Frontend::Overview'}->{'Preview'} =  {
  'CustomerInfo' => '0',
  'CustomerInfoMaxSize' => '18',
  'DefaultPreViewLines' => '25',
  'DefaultViewNewLine' => '90',
  'Module' => 'Kernel::Output::HTML::TicketOverviewPreview',
  'ModulePriority' => '300',
  'Name' => 'Large',
  'NameShort' => 'L',
  'OverviewMenuModules' => '1',
  'StripEmptyLines' => '0',
  'TicketActionsPerTicket' => '0'
};
```

6.57.5. Ticket::Frontend::Overview::PreviewArticleSenderTypes

Description:	Defines which article sender types should be shown in the preview of a ticket.
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	0
Required:	0
Config-Setting:	

```
$Self->{'Ticket::Frontend::Overview::PreviewArticleSenderTypes'} =
  {
  'agent' => '1',
  'customer' => '1',
  'system' => '1'
};
```

6.57.6. Ticket::Frontend::Overview::PreviewArticleLimit

Description:	Sets the count of articles visible in preview mode of ticket overviews.
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	1
Config-Setting:	

```
$Self->{'Ticket::Frontend::Overview::PreviewArticleLimit'} =  '5';
```

6.57.7. Ticket::Frontend::Overview::PreviewArticleTypeExpanded

Description:	Defines wich article type should be expanded when entering the overview. If nothing defined, latest article will be expanded.
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::Overview::PreviewArticleTypeExpanded'} = '';`

6.57.8. Ticket::Frontend::OverviewSmall###DynamicField

Description:	Dynamic fields shown in the ticket small format overview screen of the agent interface. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default.
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::OverviewSmall'}->{'DynamicField'} = {};`

6.57.9. Ticket::Frontend::OverviewMedium###DynamicField

Description:	Dynamic fields shown in the ticket medium format overview screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled.
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::OverviewMedium'}->{'DynamicField'} = {};`

6.57.10. Ticket::Frontend::OverviewPreview###DynamicField

Description:	Dynamic fields shown in the ticket preview format overview screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled.
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::OverviewPreview'}->{'DynamicField'} = {};`

6.58. Frontend::Agent::ToolBarModule

6.58.1. Frontend::ToolBarModule###1-Ticket::AgentTicketQueue

Description:	Toolbar Item for a shortcut.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0

Config-Setting:	<pre>$Self->{'Frontend::ToolBarModule'}->{'1-Ticket::AgentTicketQueue'} = {
	'AccessKey' => 'q',
	'Action' => 'AgentTicketQueue',
	'CssClass' => 'QueueView',
	'Icon' => 'icon-folder-close',
	'Link' => 'Action=AgentTicketQueue',
	'Module' => 'Kernel::Output::HTML::ToolBarLink',
	'Name' => 'Queue view',
	'Priority' => '1010010'
	};</pre>

6.58.2. Frontend::ToolBarModule###2-Ticket::AgentTicketStatus

Description:	Toolbar Item for a shortcut.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0

Config-Setting:	<pre>$Self->{'Frontend::ToolBarModule'}->{'2-Ticket::AgentTicketStatus'} = {
	'AccessKey' => 'o',
	'Action' => 'AgentTicketStatusView',
	'CssClass' => 'StatusView',
	'Icon' => 'icon-list-ol',
	'Link' => 'Action=AgentTicketStatusView',
	'Module' => 'Kernel::Output::HTML::ToolBarLink',
	'Name' => 'Status view',
	'Priority' => '1010020'
	};</pre>

6.58.3. Frontend::ToolBarModule###3-Ticket::AgentTicketEscalation

Description:	Toolbar Item for a shortcut.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0

Config-Setting:	<pre>$Self->{'Frontend::ToolBarModule'}->{'3-Ticket::AgentTicketEscalation'} = {
	'AccessKey' => 'w',
	'Action' => 'AgentTicketEscalationView',</pre>

```
  'CssClass' => 'EscalationView',
  'Icon' => 'icon-exclamation',
  'Link' => 'Action=AgentTicketEscalationView',
  'Module' => 'Kernel::Output::HTML::ToolBarLink',
  'Name' => 'Escalation view',
  'Priority' => '1010030'
};
```

6.58.4. Frontend::ToolBarModule###4-Ticket::AgentTicketPhone

Description:	Toolbar Item for a shortcut.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Frontend::ToolBarModule'}->{'4-Ticket::AgentTicketPhone'} = {`
	`'AccessKey' => 'l',`
	`'Action' => 'AgentTicketPhone',`
	`'CssClass' => 'PhoneTicket',`
	`'Icon' => 'icon-phone',`
	`'Link' => 'Action=AgentTicketPhone',`
	`'Module' => 'Kernel::Output::HTML::ToolBarLink',`
	`'Name' => 'New phone ticket',`
	`'Priority' => '1020010'`
	`};`

6.58.5. Frontend::ToolBarModule###5-Ticket::AgentTicketEmail

Description:	Toolbar Item for a shortcut.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Frontend::ToolBarModule'}->{'5-Ticket::AgentTicketEmail'} = {`
	`'AccessKey' => 'e',`
	`'Action' => 'AgentTicketEmail',`
	`'CssClass' => 'EmailTicket',`
	`'Icon' => 'icon-envelope',`
	`'Link' => 'Action=AgentTicketEmail',`
	`'Module' => 'Kernel::Output::HTML::ToolBarLink',`
	`'Name' => 'New email ticket',`
	`'Priority' => '1020020'`
	`};`

6.58.6. Frontend::ToolBarModule###6-Ticket::AgentTicketProcess

Description:	Toolbar Item for a shortcut.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0

Config-Setting:	<pre>$Self->{'Frontend::ToolBarModule'}->{'6-Ticket::AgentTicketProcess'} = {
	'AccessKey' => 'p',
	'Action' => 'AgentTicketProcess',
	'CssClass' => 'ProcessTicket',
	'Icon' => 'icon-th-large',
	'Link' => 'Action=AgentTicketProcess',
	'Module' => 'Kernel::Output::HTML::ToolBarLink',
	'Name' => 'New process ticket',
	'Priority' => '1020030'
	};</pre>

6.58.7. Frontend::ToolBarModule###6-Ticket::TicketResponsible

Description:	Agent interface notification module to see the number of tickets an agent is responsible for.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Frontend::ToolBarModule'}->{'6-Ticket::TicketResponsible'} = {
	'CssClass' => 'Responsible',
	'CssClassNew' => 'Responsible New',
	'CssClassReached' => 'Responsible Reached',
	'Icon' => 'icon-user',
	'IconNew' => 'icon-user',
	'IconReached' => 'icon-user',
	'Module' => 'Kernel::Output::HTML::ToolBarTicketResponsible',
	'Priority' => '1030010'
	};</pre>

6.58.8. Frontend::ToolBarModule###7-Ticket::TicketWatcher

Description:	Agent interface notification module to see the number of watched tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Frontend::ToolBarModule'}->{'7-Ticket::TicketWatcher'} = {
	'CssClass' => 'Watcher',
	'CssClassNew' => 'Watcher New',
	'CssClassReached' => 'Watcher Reached',
	'Icon' => 'icon-eye-open',
	'IconNew' => 'icon-eye-open',
	'IconReached' => 'icon-eye-open',
	'Module' => 'Kernel::Output::HTML::ToolBarTicketWatcher',
	'Priority' => '1030020'
	};</pre>

6.58.9. Frontend::ToolBarModule###8-Ticket::TicketLocked

Description:	Agent interface notification module to check the used charset.

Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'Frontend::ToolBarModule'}->{'8-Ticket::TicketLocked'} = {
	  'CssClass' => 'Locked',
	  'CssClassNew' => 'Locked New',
	  'CssClassReached' => 'Locked Reached',
	  'Icon' => 'icon-lock',
	  'IconNew' => 'icon-lock',
	  'IconReached' => 'icon-lock',
	  'Module' => 'Kernel::Output::HTML::ToolBarTicketLocked',
	  'Priority' => '1030030'
	};</pre>

6.58.10. Frontend::ToolBarModule###9-Ticket::TicketSearchProfile

Description:	Agent interface module to access search profiles via nav bar.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0

Config-Setting:	<pre>$Self->{'Frontend::ToolBarModule'}->{'9-Ticket::TicketSearchProfile'} = {
	  'Block' => 'ToolBarSearchProfile',
	  'Description' => 'Search-Template',
	  'MaxWidth' => '40',
	  'Module' => 'Kernel::Output::HTML::ToolBarTicketSearchProfile',
	  'Name' => 'Search-Template',
	  'Priority' => '1990010'
	};</pre>

6.58.11. Frontend::ToolBarModule###10-Ticket::TicketSearchFulltext

Description:	Agent interface module to access fulltext search via nav bar.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0

Config-Setting:	<pre>$Self->{'Frontend::ToolBarModule'}->{'10-Ticket::TicketSearchFulltext'} = {
	  'Block' => 'ToolBarSearchFulltext',
	  'CSS' => 'Core.Agent.Toolbar.FulltextSearch.css',
	  'Description' => 'Fulltext search',
	  'Module' => 'Kernel::Output::HTML::ToolBarGeneric',
	  'Name' => 'Fulltext-Search',
	  'Priority' => '1990020',
	  'Size' => '10'
	};</pre>

6.58.12. Frontend::ToolBarModule###11-CICSearchCustomerID

Description:	Agent interface module to access CIC search via nav bar.

Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Frontend::ToolBarModule'}->{'11-CICSearchCustomerID'} = { 'Block' => 'ToolBarCICSearchCustomerID', 'CSS' => 'Core.Agent.Toolbar.CICSearch.css', 'Description' => 'CustomerID Search', 'Module' => 'Kernel::Output::HTML::ToolBarGeneric', 'Name' => 'CustomerID search', 'Priority' => '1990030', 'Size' => '10' };`

6.58.13. Frontend::ToolBarModule###11-CICSearchCustomerUser

Description:	Agent interface module to access CIC search via nav bar.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Frontend::ToolBarModule'}->{'11-CICSearchCustomerUser'} = { 'Block' => 'ToolBarCICSearchCustomerUser', 'CSS' => 'Core.Agent.Toolbar.CICSearch.css', 'Description' => 'CustomerUser Search', 'Module' => 'Kernel::Output::HTML::ToolBarGeneric', 'Name' => 'Customer user search', 'Priority' => '1990040', 'Size' => '10' };`

6.59. Frontend::Customer

6.59.1. Ticket::Frontend::CustomerDisableCompanyTicketAccess

Description:	This option will deny the access to customer company tickets, which are not created by the customer user.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerDisableCompanyTicketAccess'} = '0';`

6.59.2. Ticket::Frontend::CustomerTicketOverviewCustomEmptyText

Description:	Custom text for the page shown to customers that have no tickets yet.

Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketOverviewCustomEmptyText'} = {`
	`'Button' => 'Create your first ticket',`
	`'Text' => 'Please click the button below to create your first ticket.',`
	`'Title' => 'Welcome!'`
	`};`

6.59.3. Frontend::CustomerUser::Item###15-OpenTickets

Description:	Customer item (icon) which shows the open tickets of this customer as info block. Setting CustomerUserLogin to 1 searches for tickets based on login name rather than CustomerID.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::CustomerUser::Item'}->{'15-OpenTickets'} = {`
	`'Action' => 'AgentTicketSearch',`
	`'Attributes' => 'StateType=Open;',`
	`'CSS' => 'Core.Agent.CustomerUser.OpenTicket.css',`
	`'CSSClassNoOpenTicket' => 'NoOpenTicket',`
	`'CSSClassOpenTicket' => 'OpenTicket',`
	`'CustomerUserLogin' => '0',`
	`'Module' => 'Kernel::Output::HTML::CustomerUserGenericTicket',`
	`'Subaction' => 'Search',`
	`'Target' => '_blank',`
	`'Text' => 'Open tickets'`
	`};`

6.59.4. Frontend::CustomerUser::Item###16-OpenTicketsForCustomerUserLogin

Description:	Customer item (icon) which shows the open tickets of this customer as info block. Setting CustomerUserLogin to 1 searches for tickets based on login name rather than CustomerID.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Frontend::CustomerUser::Item'}->{'16-OpenTicketsForCustomerUserLogin'} = {`
	`'Action' => 'AgentTicketSearch',`
	`'Attributes' => 'StateType=Open;',`
	`'CSS' => 'Core.Agent.CustomerUser.OpenTicket.css',`
	`'CSSClassNoOpenTicket' => 'NoOpenTicket',`
	`'CSSClassOpenTicket' => 'OpenTicket',`
	`'CustomerUserLogin' => '1',`
	`'Module' => 'Kernel::Output::HTML::CustomerUserGenericTicket',`
	`'Subaction' => 'Search',`
	`'Target' => '_blank',`

	'Text' => 'Open tickets of customer' };

6.59.5. Frontend::CustomerUser::Item###17-ClosedTickets

Description:	Customer item (icon) which shows the closed tickets of this customer as info block. Setting CustomerUserLogin to 1 searches for tickets based on login name rather than CustomerID.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	0
Required:	0

Config-Setting:	$Self->{'Frontend::CustomerUser::Item'}->{'17-ClosedTickets'} = {
	'Action' => 'AgentTicketSearch',
	'Attributes' => 'StateType=Closed;',
	'CSS' => 'Core.Agent.CustomerUser.OpenTicket.css',
	'CSSClassNoOpenTicket' => 'NoOpenTicket',
	'CSSClassOpenTicket' => 'OpenTicket',
	'CustomerUserLogin' => '0',
	'Module' => 'Kernel::Output::HTML::CustomerUserGenericTicket',
	'Subaction' => 'Search',
	'Target' => '_blank',
	'Text' => 'Closed tickets'
	};

6.59.6. Frontend::CustomerUser::Item###18-ClosedTicketsForCustomerUserLogin

Description:	Customer item (icon) which shows the closed tickets of this customer as info block. Setting CustomerUserLogin to 1 searches for tickets based on login name rather than CustomerID.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	0
Required:	0

Config-Setting:	$Self->{'Frontend::CustomerUser::Item'}->{'18-ClosedTicketsForCustomerUserLogin'} = {
	'Action' => 'AgentTicketSearch',
	'Attributes' => 'StateType=Closed;',
	'CSS' => 'Core.Agent.CustomerUser.OpenTicket.css',
	'CSSClassNoOpenTicket' => 'NoOpenTicket',
	'CSSClassOpenTicket' => 'OpenTicket',
	'CustomerUserLogin' => '1',
	'Module' => 'Kernel::Output::HTML::CustomerUserGenericTicket',
	'Subaction' => 'Search',
	'Target' => '_blank',
	'Text' => 'Closed tickets of customer'
	};

6.59.7. CustomerFrontend::CommonObject###QueueObject

Description:	Path of the file that stores all the settings for the QueueObject object for the customer interface.
Group:	Ticket

SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerFrontend::CommonObject'}->{'QueueObject'} = 'Kernel::System::Queue';`

6.59.8. CustomerFrontend::CommonObject###TicketObject

Description:	Path of the file that stores all the settings for the TicketObject for the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerFrontend::CommonObject'}->{'TicketObject'} = 'Kernel::System::Ticket';`

6.59.9. CustomerFrontend::CommonParam###Action

Description:	Defines the default used Frontend-Module if no Action parameter given in the url on the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerFrontend::CommonParam'}->{'Action'} = 'CustomerTicketOverview';`

6.59.10. CustomerFrontend::CommonParam###TicketID

Description:	Default ticket ID used by the system in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerFrontend::CommonParam'}->{'TicketID'} =  '';`

6.60. Frontend::Customer::ModuleMetaHead

6.60.1. CustomerFrontend::HeaderMetaModule###2-TicketSearch

Description:	Module to generate html OpenSearch profile for short ticket search in the customer interface.
Group:	Ticket

SubGroup:	Frontend::Customer::ModuleMetaHead
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerFrontend::HeaderMetaModule'}->{'2-TicketSearch'} = { 'Action' => 'CustomerTicketSearch', 'Module' => 'Kernel::Output::HTML::CustomerHeaderMetaTicketSearch' };`

6.61. Frontend::Customer::ModuleRegistration

6.61.1. CustomerFrontend::Module###CustomerTicketOverview

Description:	Frontend module registration for the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerFrontend::Module'}->{'CustomerTicketOverview'} = { 'Description' => 'Overview of customer tickets', 'NavBar' => [{ 'AccessKey' => 't', 'Block' => '', 'Description' => 'Tickets', 'Link' => 'Action=CustomerTicketOverview;Subaction=MyTickets', 'LinkOption' => '', 'Name' => 'Tickets', 'NavBar' => 'Ticket', 'Prio' => '100', 'Type' => 'Menu' }, { 'AccessKey' => 'm', 'Block' => '', 'Description' => 'My Tickets', 'Link' => 'Action=CustomerTicketOverview;Subaction=MyTickets', 'LinkOption' => '', 'Name' => 'My Tickets', 'NavBar' => 'Ticket', 'Prio' => '110', 'Type' => 'Submenu' }, { 'AccessKey' => 'c', 'Block' => '', 'Description' => 'Company Tickets', 'Link' => 'Action=CustomerTicketOverview;Subaction=CompanyTickets', 'LinkOption' => '', 'Name' => 'Company Tickets', 'NavBar' => 'Ticket', 'Prio' => '120', 'Type' => 'Submenu' }], 'NavBarName' => 'Ticket', 'Title' => 'Overview' };`

6.61.2. CustomerFrontend::Module###CustomerTicketMessage

Description:	Frontend module registration for the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'CustomerFrontend::Module'}->{'CustomerTicketMessage'} = {
	  'Description' => 'Create tickets',
	  'NavBar' => [
	    {
	      'AccessKey' => 'n',
	      'Block' => '',
	      'Description' => 'Create new Ticket',
	      'Link' => 'Action=CustomerTicketMessage',
	      'LinkOption' => '',
	      'Name' => 'New Ticket',
	      'NavBar' => 'Ticket',
	      'Prio' => '100',
	      'Type' => 'Submenu'
	    }
	  ],
	  'NavBarName' => 'Ticket',
	  'Title' => 'New Ticket'
	};</pre>

6.61.3. CustomerFrontend::Module###CustomerTicketZoom

Description:	Frontend module registration for the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'CustomerFrontend::Module'}->{'CustomerTicketZoom'} = {
	  'Description' => 'Ticket zoom view',
	  'Loader' => {
	    'JavaScript' => [
	      'Core.Customer.TicketZoom.js',
	      'Core.UI.Popup.js'
	    ]
	  },
	  'NavBarName' => 'Ticket',
	  'Title' => 'Zoom'
	};</pre>

6.61.4. CustomerFrontend::Module###CustomerTicketPrint

Description:	Frontend module registration for the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'CustomerFrontend::Module'}->{'CustomerTicketPrint'} = { 'Description' => 'Customer Ticket Print Module', 'NavBarName' => '', 'Title' => 'Print' };`

6.61.5. CustomerFrontend::Module###CustomerTicketAttachment

Description:	Frontend module registration for the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerFrontend::Module'}->{'CustomerTicketAttachment'} = { 'Description' => 'To download attachments', 'NavBarName' => '', 'Title' => '' };`

6.61.6. CustomerFrontend::Module###CustomerTicketSearch

Description:	Frontend module registration for the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerFrontend::Module'}->{'CustomerTicketSearch'} = { 'Description' => 'Customer ticket search', 'NavBar' => [{ 'AccessKey' => 's', 'Block' => '', 'Description' => 'Search', 'Link' => 'Action=CustomerTicketSearch', 'LinkOption' => '', 'Name' => 'Search', 'NavBar' => 'Ticket', 'Prio' => '300', 'Type' => 'Submenu' }], 'NavBarName' => 'Ticket', 'Title' => 'Search' };`

6.62. Frontend::Customer::Preferences

6.62.1. CustomerPreferencesGroups###ShownTickets

Description:	Defines all the parameters for the ShownTickets object in the customer preferences of the customer interface.
Group:	Ticket

SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'CustomerPreferencesGroups'}->{'ShownTickets'} = {
	'Active' => '1',
	'Column' => 'User Profile',
	'Data' => {
	'15' => '15',
	'20' => '20',
	'25' => '25',
	'30' => '30'
	},
	'DataSelected' => '25',
	'Key' => 'Tickets per page',
	'Label' => 'Number of displayed tickets',
	'Module' => 'Kernel::Output::HTML::PreferencesGeneric',
	'PrefKey' => 'UserShowTickets',
	'Prio' => '4000'
	};</pre>

6.62.2. CustomerPreferencesGroups###RefreshTime

Description:	Defines all the parameters for the RefreshTime object in the customer preferences of the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0

Config-Setting:	<pre>$Self->{'CustomerPreferencesGroups'}->{'RefreshTime'} = {
	'Active' => '1',
	'Column' => 'User Profile',
	'Data' => {
	'' => 'off',
	'10' => '10 minutes',
	'15' => '15 minutes',
	'2' => ' 2 minutes',
	'5' => ' 5 minutes',
	'7' => ' 7 minutes'
	},
	'DataSelected' => '',
	'Key' => 'Refresh interval',
	'Label' => 'Ticket overview',
	'Module' => 'Kernel::Output::HTML::PreferencesGeneric',
	'PrefKey' => 'UserRefreshTime',
	'Prio' => '4000'
	};</pre>

6.63. Frontend::Customer::Ticket::ViewNew

6.63.1. Ticket::Frontend::CustomerTicketMessage###NextScreenAfterN

Description:	Determines the next screen after new customer ticket in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'NextScreenAfterNewTicket'} = 'CustomerTicketOverview';`

6.63.2. Ticket::Frontend::CustomerTicketMessage###Priority

Description:	Allows customers to set the ticket priority in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'Priority'} = '1';`

6.63.3. Ticket::Frontend::CustomerTicketMessage###PriorityDefault

Description:	Defines the default priority of new customer tickets in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'PriorityDefault'} = '3 normal';`

6.63.4. Ticket::Frontend::CustomerTicketMessage###Queue

Description:	Allows customers to set the ticket queue in the customer interface. If this is set to 'No', QueueDefault should be configured.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'Queue'} = '1';`

6.63.5. Ticket::Frontend::CustomerTicketMessage###QueueDefault

Description:	Defines the default queue for new customer tickets in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	0

Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'QueueDefault'} = 'Postmaster';`

6.63.6. Ticket::Frontend::CustomerTicketMessage###TicketType

Description:	Allows customers to set the ticket type in the customer interface. If this is set to 'No', TicketTypeDefault should be configured.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'TicketType'} = '1';`

6.63.7. Ticket::Frontend::CustomerTicketMessage###TicketTypeDefault

Description:	Defines the default ticket type for new customer tickets in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'TicketTypeDefault'} = 'default';`

6.63.8. Ticket::Frontend::CustomerTicketMessage###Service

Description:	Allows customers to set the ticket service in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'Service'} = '1';`

6.63.9. Ticket::Frontend::CustomerTicketMessage###SLA

Description:	Allows customers to set the ticket SLA in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1

Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'SLA'} = '1';`

6.63.10. Ticket::Frontend::CustomerTicketMessage###ServiceMandatory

Description:	Sets if service must be selected by the customer.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'ServiceMandatory'} = '0';`

6.63.11. Ticket::Frontend::CustomerTicketMessage###SLAMandatory

Description:	Sets if SLA must be selected by the customer.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'SLAMandatory'} = '0';`

6.63.12. Ticket::Frontend::CustomerTicketMessage###StateDefault

Description:	Defines the default state of new customer tickets in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'StateDefault'} = 'new';`

6.63.13. Ticket::Frontend::CustomerTicketMessage###ArticleType

Description:	Defines the default type for article in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'ArticleType'} = 'webrequest';`

6.63.14. Ticket::Frontend::CustomerTicketMessage###SenderType

Description:	Sender type for new tickets from the customer inteface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'SenderType'} = 'customer';`

6.63.15. Ticket::Frontend::CustomerTicketMessage###HistoryType

Description:	Defines the default history type in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'HistoryType'} = 'WebRequestCustomer';`

6.63.16. Ticket::Frontend::CustomerTicketMessage###HistoryComment

Description:	Comment for new history entries in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'HistoryComment'} = '';`

6.63.17. CustomerPanelSelectionType

Description:	Defines the receipent target of the tickets ("Queue" shows all queues, "SystemAddress" displays all system addresses) in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerPanelSelectionType'} = 'Queue';`

6.63.18. CustomerPanelSelectionString

Description:	Determines the strings that will be shown as receipent (To:) of the ticket in the customer interface. For Queue as

	CustomerPanelSelectionType, "<Queue>" shows the names of the queues, and for SystemAddress, "<Realname> <<Email>>" shows the name and email of the receipent.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerPanelSelectionString'} = '<Queue>';`

6.63.19. CustomerPanelOwnSelection

Description:	Determines which queues will be valid for ticket's recepients in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	0
Required:	0
Config-Setting:	`$Self->{'CustomerPanelOwnSelection'} = { 'Junk' => 'First Queue', 'Misc' => 'Second Queue' };`

6.63.20. CustomerPanel::NewTicketQueueSelectionModule

Description:	Module for To-selection in new ticket screen in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerPanel::NewTicketQueueSelectionModule'} = 'Kernel::Output::HTML::CustomerNewTicketQueueSelectionGeneric';`

6.63.21. Ticket::Frontend::CustomerTicketMessage###DynamicField

Description:	Dynamic fields options shown in the ticket message screen of the customer interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required. NOTE. If you want to display these fields also in the ticket zoom of the customer interface, you have to enable them in CustomerTicketZoom###DynamicField.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'DynamicField'} = {};`

6.64. Frontend::Customer::Ticket::ViewPrint

6.64.1. Ticket::Frontend::CustomerTicketPrint###DynamicField

Description:	Dynamic fields shown in the ticket print screen of the customer interface. Possible settings: 0 = Disabled, 1 = Enabled.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewPrint
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketPrint'}->{'DynamicField'} = {};`

6.65. Frontend::Customer::Ticket::ViewSearch

6.65.1. Ticket::CustomerTicketSearch::SearchLimit

Description:	Maximum number of tickets to be displayed in the result of a search in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::CustomerTicketSearch::SearchLimit'} = '5000';`

6.65.2. Ticket::CustomerTicketSearch::SearchPageShown

Description:	Number of tickets to be displayed in each page of a search result in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::CustomerTicketSearch::SearchPageShown'} = '40';`

6.65.3. Ticket::CustomerTicketSearch::SortBy::Default

Description:	Defines the default ticket attribute for ticket sorting in a ticket search of the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::CustomerTicketSearch::SortBy::Default'} = 'Age';`

6.65.4. Ticket::CustomerTicketSearch::Order::Default

Description:	Defines the default ticket order of a search result in the customer interface. Up: oldest on top. Down: latest on top.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::CustomerTicketSearch::Order::Default'} = 'Down';`

6.65.5. Ticket::Frontend::CustomerTicketSearch###ExtendedSearchCor

Description:	Allows extended search conditions in ticket search of the customer interface. With this feature you can search w. g. with this kind of conditions like "(key1&&key2)" or "(key1||key2)".
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketSearch'}->{'ExtendedSearchCondition'} = '1';`

6.65.6. Ticket::Frontend::CustomerTicketSearch###SearchCSVData

Description:	Data used to export the search result in CSV format.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	1

Config-Setting:	<pre>$Self->{'Ticket::Frontend::CustomerTicketSearch'}->{'SearchCSVData'} = [
	  'TicketNumber',
	  'Age',
	  'Created',
	  'Closed',
	  'State',
	  'Priority',
	  'Lock',
	  'CustomerID',
	  'CustomerName',
	  'From',
	  'Subject'
	];</pre>

6.65.7. Ticket::Frontend::CustomerTicketSearch###DynamicField

Description:	Dynamic fields shown in the ticket search screen of the customer interface. Possible settings: 0 = Disabled, 1 = Enabled.
Group:	Ticket

SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketSearch'}->{'DynamicField'} = {};`

6.65.8. Ticket::Frontend::CustomerTicketSearch###SearchOverviewDyn

Description:	Dynamic fields shown in the ticket search overview results screen of the customer interface. Possible settings: 0 = Disabled, 1 = Enabled.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketSearch'}->{'SearchOverviewDynamicField'} = {};`

6.65.9. Ticket::Frontend::CustomerTicketSearch###SearchCSVDynamic

Description:	Dynamic Fields used to export the search result in CSV format.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketSearch'}->{'SearchCSVDynamicField'} = {};`

6.66. Frontend::Customer::Ticket::ViewZoom

6.66.1. Ticket::Frontend::CustomerTicketZoom###NextScreenAfterFollo

Description:	Determines the next screen after the follow up screen of a zoomed ticket in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'NextScreenAfterFollowUp'} = 'CustomerTicketOverview';`

6.66.2. Ticket::Frontend::CustomerTicketZoom###ArticleType

Description:	Defines the default type of the note in the ticket zoom screen of the customer interface.

Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'ArticleType'} = 'webrequest';`

6.66.3. Ticket::Frontend::CustomerTicketZoom###SenderType

Description:	Defines the default sender type for tickets in the ticket zoom screen of the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'SenderType'} = 'customer';`

6.66.4. Ticket::Frontend::CustomerTicketZoom###HistoryType

Description:	Defines the history type for the ticket zoom action, which gets used for ticket history in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'HistoryType'} = 'FollowUp';`

6.66.5. Ticket::Frontend::CustomerTicketZoom###HistoryComment

Description:	Defines the history comment for the ticket zoom action, which gets used for ticket history in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'HistoryComment'} = '';`

6.66.6. Ticket::Frontend::CustomerTicketZoom###Priority

Description:	Allows customers to change the ticket priority in the customer interface.

Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'Priority'} = '1';`

6.66.7. Ticket::Frontend::CustomerTicketZoom###PriorityDefault

Description:	Defines the default priority of follow up customer tickets in the ticket zoom screen in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'PriorityDefault'} = '3 normal';`

6.66.8. Ticket::Frontend::CustomerTicketZoom###State

Description:	Allows choosing the next compose state for customer tickets in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'State'} = '1';`

6.66.9. Ticket::Frontend::CustomerTicketZoom###StateDefault

Description:	Defines the default next state for a ticket after customer follow up in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'StateDefault'} = 'open';`

6.66.10. Ticket::Frontend::CustomerTicketZoom###StateType

Description:	Defines the next possible states for customer tickets in the customer interface.

Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'StateType'} = [ 'open', 'closed' ];`

6.66.11. Ticket::Frontend::CustomerTicketZoom###AttributesView

Description:	Shows the activated ticket attributes in the customer interface (0 = Disabled and 1 = Enabled).
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'AttributesView'} = { 'Owner' => '0', 'Priority' => '1', 'Queue' => '1', 'Responsible' => '0', 'SLA' => '0', 'Service' => '0', 'State' => '1', 'Type' => '0' };`

6.66.12. Ticket::Frontend::CustomerTicketZoom###DynamicField

Description:	Dynamic fields shown in the ticket zoom screen of the customer interface. Possible settings: 0 = Disabled, 1 = Enabled.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'DynamicField'} = {};`

6.66.13. Ticket::Frontend::CustomerTicketZoom###FollowUpDynamicFie

Description:	Dynamic fields options shown in the ticket reply section in the ticket zoom screen of the customer interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1

Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'FollowUpDynamicField'} = {};`

6.67. Frontend::Customer::TicketOverview

6.67.1. Ticket::Frontend::CustomerTicketOverviewSortable

Description:	Controls if customers have the ability to sort their tickets.
Group:	Ticket
SubGroup:	Frontend::Customer::TicketOverview
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketOverviewSortable'} = '';`

6.67.2. Ticket::Frontend::CustomerTicketOverview###ColumnHeader

Description:	Shows either the last customer article's subject or the ticket title in the small format overview.
Group:	Ticket
SubGroup:	Frontend::Customer::TicketOverview
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketOverview'}->{'ColumnHeader'} = 'TicketTitle';`

6.67.3. Ticket::Frontend::CustomerTicketOverview###Owner

Description:	Show the current owner in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::TicketOverview
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketOverview'}->{'Owner'} = '0';`

6.67.4. Ticket::Frontend::CustomerTicketOverview###Queue

Description:	Show the current queue in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::TicketOverview
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketOverview'}->{'Queue'} = '0';`

6.67.5. Ticket::Frontend::CustomerTicketOverview###DynamicField

Description:	Dynamic fields shown in the ticket overview screen of the customer interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Customer::TicketOverview
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketOverview'}->{'DynamicField'} = {};`

6.68. Frontend::Queue::Preferences

6.68.1. QueuePreferences###Comment2

Description:	Parameters of the example queue attribute Comment2.
Group:	Ticket
SubGroup:	Frontend::Queue::Preferences
Valid:	0
Required:	0
Config-Setting:	`$Self->{'QueuePreferences'}->{'Comment2'} = { 'Block' => 'TextArea', 'Cols' => '50', 'Desc' => 'Define the queue comment 2.', 'Label' => 'Comment2', 'Module' => 'Kernel::Output::HTML::QueuePreferencesGeneric', 'PrefKey' => 'Comment2', 'Rows' => '5' };`

6.69. Frontend::SLA::Preferences

6.69.1. SLAPreferences###Comment2

Description:	Parameters of the example SLA attribute Comment2.
Group:	Ticket
SubGroup:	Frontend::SLA::Preferences
Valid:	0
Required:	0
Config-Setting:	`$Self->{'SLAPreferences'}->{'Comment2'} = { 'Block' => 'TextArea', 'Cols' => '50', 'Desc' => 'Define the sla comment 2.', 'Label' => 'Comment2', 'Module' => 'Kernel::Output::HTML::SLAPreferencesGeneric', 'PrefKey' => 'Comment2', 'Rows' => '5' };`

6.70. Frontend::Service::Preferences

6.70.1. ServicePreferences###Comment2

Description:	Parameters of the example service attribute Comment2.
Group:	Ticket
SubGroup:	Frontend::Service::Preferences
Valid:	0
Required:	0

Config-Setting:	<pre>$Self->{'ServicePreferences'}->{'Comment2'} = {
	  'Block' => 'TextArea',
	  'Cols' => '50',
	  'Desc' => 'Define the service comment 2.',
	  'Label' => 'Comment2',
	  'Module' => 'Kernel::Output::HTML::ServicePreferencesGeneric',
	  'PrefKey' => 'Comment2',
	  'Rows' => '5'
	};</pre>

付録C 敬意と賞賛

OTRSは、オープンソース・プロジェクトです。多くの方々の援助やサポートに感謝します。次の挙げる人名リストは不完全に違いないことをお詫びします。もしこのリストにお名前がないようでしたらお知らせ下さい。

以下の方々は特にプロジェクトを推進してくれ現在もなおアクティブなサポーターです。

- Robert Kehl (以下敬称略) は、Win32-installer for the 1.x releasesを作成してくれました。Robertに感謝！
- Torsten Wernerは、Debian-installer for OTRSをメンテナンスしてくれています。OTRSをDebianコミュニティでも利用できるようにしてくれてありがとう、Torsten！
- Nils Jeppe (mirror Hamburg, Germany), Bryan Fullerton (mirror Toronto, Canada), Eberhard Mönkeberg (mirror Göttingen, Germany), Timo Dreger (mirror Düsseldorf, Germany) and Netmonic (mirror Vienna, Austria)は、FTPサーバのミラーリングをしています。常にOTRSを即座にダウンロードできるように支援してくれてありがとう！
- Anja Schneiderはマニュアルの翻訳、ドイツ語テキストの訂正や改訂を支援してくれています。Anja、あなたの支援と忍耐に感謝します！
- 私達はOTRSのシステム改善、パッチあるいはバグフィックス用のメーリングリストから多くのアイデアを頂いています。そこでは常に全てのユーザが素晴らしいサポートを享受できます。メーリングリストご利用の全ての方々の活発なご支援に深く感謝します。

また次の方々にも深く感謝します。

- Martin Scherbaum
- Carsten Gross
- Harald Müller
- Stefan Schmidt
- Milisav Radmanic
- Uli Hecht
- Norman Walsh
- Heiko Baumann
- Atif Ghaffar
- Pablo Ruiz Garcia
- Dan Rau
- Christoph Kaulich
- Mark Jackson
- Diane Shieh
- Bernard Choppy
- Carl Bailey
- Phil Davis

- Edwin D. Vinas
- Lars Müller
- Vladimir Gerdjikov
- Fred van Dijk
- Sebastien Guilbaud
- Wiktor Wodecki
- Arnold Ligtvoet
- Antti Kämäräinen
- Nicolas Goralski
- Gilberto Cezar de Almeida
- Jorge Becerra
- Eddie Urenda
- Stella Power
- Andreas Haase
- Reiner Keller
- Covert Jake
- Moshe Leibovitch
- Bjärn Jacke
- Remo Catelotti
- Alfons Obermeyer
- Michael Rech
- Danie Theron
- Richard Paradies
- Art Powell

付録D GNUフリー文書許諾契約書

Version 1.1, March 2000

> Copyright (C) 2000 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA このライセンス・ドキュメントの逐語的な原稿は誰でもコピーするおよび配布することを許可します。しかし変更は許可しません。

0. はじめに

この利用許諾契約書の目的は、この契約書が適用されるマニュアルや教科書、 その他機能本位で実用的な文書を(無料ではなく)自由という意味で「フリー」 とすること、すなわち、改変の有無あるいは目的の営利非営利を問わず、文書 を複製し再頒布する自由をすべての人々に効果的に保証することです。加えて この契約書により、著者や出版者が自分たちの著作物に対して相応の敬意と賞賛を得る手段も保護されます。また、他人が行った改変に対して責任を負わず に済むようになります。

この利用許諾契約書は「コピーレフト」的なライセンスの一つであり、この契 約書が適用された文書から派生した著作物は、それ自身もまた原本と同じ意味 でフリーでなければなりません。この契約書は、フリーソフトウェアのために 設計されたコピーレフトなライセンスであるGNU一般公衆使用許諾契約書を補足 するものです。

この利用許諾契約書は、フリーソフトウェア用のマニュアルに適用することを 目的として書かれました。フリーソフトウェアはフリーな文書を必要としてお り、フリーなプログラムはそのソフトウェアが保証するのと同じ自由を提供す るマニュアルと共に頒布されるべきだからです。しかし、この契約書の適用範 囲はソフトウェアのマニュアルに留まりません。対象となる著作物において扱われる主題が何であれ、あるいはそれが印刷された書籍として出版されるか否 かに関わらず、この契約書は文字で書かれたいかなる著作物にも適用すること が可能です。私たちとしては、主にこの契約書を解説や参照を目的とする著作 物に適用することをお勧めします。

1. この利用許諾契約書の適用範囲と用語の定義

著作物がこの利用許諾契約書の定める条件の下で頒布される旨の告知を、著作 権者がその中に書いたすべてのマニュアルあるいはその他の著作物は、この契約書の適用対象となります。以下において、「『文書』(Document)」 とはそのような告知が記載されたマニュアルないし著作物すべてを指します。公衆の一員ならば誰でも契約の当事者となることができ、この契約書中では「あなた」と表現されます。

『文書』の「改変版 (Modified Version)」とは、一字一句忠実に複製したか、 あるいは改変や他言語への翻訳を行ったかどうかに関わらず、その『文書』の 全体あるいは一部分を含む著作物すべてを意味します。

「補遺部分 (Secondary Section)」とは、文書中でその旨指定された補遺 ないし本文に先だって前付けとして置かれる一部分であり、文書の出版者 あるいは著者と、文書全体の主題 (あるいはそれに関連する事柄)との関係のみを論じ、全体としての主題の範疇に直接属する内容を全く含まないものです (たとえば、文書の一部が数学の教科書だった場合、補遺部分では数学について何も解説してはなりません)。補遺部分で扱われる関係は、その主題あるいは関連する事柄との歴史的なつながりのことかも知れないし、それらに関する法的、商業的、哲学的、倫理的、あるいは政治的立場についてかも知れません。

「変更不可部分 (Invariant Sections)」とは補遺部分の一種で、それらが変更 不可部分であることが、文書をこの利用許諾契約書の下で発表する旨述べ た告知中においてその部分の題名と共に明示されているものです。

「カバーテキスト(Cover Texts)」とは、文書がこの利用許諾契約書の指定する条件の下で発表される旨述べた告知において、「表カバーテキスト」あるいは「裏カバーテキスト」として列挙された短い文章のことを指します。

文書の「透過的」複製物とは、機械による読み取りが可能な文書の複製物のことを指します。透過的な複製物の文書形式は、その仕様が一般の人々に入手可能で、文書の内容を一般的なテキストエディタ、または(画素で構成さ　れる画像ならば)一般的なペイントプログラム、あるいは(図面ならば)いくつかの広く入手可能な製図エディタで簡単に改訂するのに適しており、なおかつテキストフォーマッタへの入力に適する(あるいはテキストフォーマッタへの入力に適する諸形式への自動的な変換に適する)ものでなければなりません。透過的なファイル形式への複製であっても、マークアップ、あるいはマークアップの不在が読者によるそれ以降の改変をわざと邪魔し阻害するように仕組まれたものは透過的であるとは見做されません。ある画像形式が、相当量のテキスト文章を表現するために使われた場合、それは透過的ではない。透過的ではない複製は「非透過的」複製と呼ばれます。

透過的複製に適した形式のサンプルとしては、マークアップを含まないプレーンな　　ASCII形式、Texinfo入力形式、LaTeX入力形式、一般に入手可能なDTDを用いたSGMLあるいはXML、または人間による改変を想定して設計された、標準に準拠したシンプルなHTMLやPostScript、PDFなどが挙げられます。透過的な画像形式の例には、PNGやXCF、JPGが含まれます。非透過な形式としては、独占的なワードプロ セッサでのみ閲覧編集できる独占的なファイル形式、普通には入手できないDTDまたは処理系を使ったSGMLやXML、ある種のワードプロセッサが生成する、出力のみを目的とした機械生成のHTMLやPostScript、PDFなどが含まれます。

「題扉 (Title Page)」とは、印刷された書籍に於いては、実際の表紙自身のみ ならず、この利用許諾契約書が表紙に掲載することを義務づける文章や図など を、読みやすい形で載せるのに必要なだけの、表紙に引き続く数ページをも意味する。表紙に類するものが無い形式で発表される著作物においては、「題扉」とは本文の始まりに先だって、その著作物の題名が最も目立つ形で現れる場所の近くに置かれる文章のことを指します。

2. 逐語的に忠実な複製

この利用許諾契約書、著作権表示、この契約書が文書に適用される旨述べた告知の三つがすべての複製物に複製され、かつあなたがこの契約書で指定されている以外のいかなる条件も追加しない限り、あなたはこの文書を、商用であるか否かを問わずいかなる形でも複製頒布することができます。あなたは、あなたが作成あるいは頒布する複製物に対して、閲覧や再複製を技術的な手法によって妨害、規制してはいけません。しかしながら、複製と引き換えに代価を得てもかまいません。あなたが相当量の複製物を頒布する際には、本契約書第3項で指定される条件にも従わなければなりません。

またあなたは、上記と同じ条件の下で、複製物を貸与したり複製物を公に開示　することができます。

3. 大量の複製

もしあなたが、文書の印刷された　(あるいは通常は印刷された表紙を持つ　媒体における)複製物を100部を超えて出版し、また文書の利用許諾告知が カバーテキストの掲載を要求している場合には、指定されたすべてのカバーテ キストを、表カバーテキストは表表紙に、裏カバーテキストは裏表紙に、はっ きりと読みやすい形で載せた表紙の中に複製物本体を綴じ込まなければなりません。また、両方の表紙において、それらの複製物の出版者としてのあなたをはっ きりとかつ読みやすい形で確認できなければなりません。表表紙では文書の完全な題名を、題名を構成するすべての語が等しく目立つようにして、視認可 能な形で示さなければなりません。それらの情報に加えて、表紙に他の文章や図 などを加えることは許可される。表紙のみを変更した複製物は、それが文書の題名を保存し上記の条件を満たす限り、ほかの点では逐語的に忠実な複製物として扱われます。

もしどちらかの表紙に要求されるカバーテキストの量が多すぎて読みやすく収 めることが不可能ならば、あなたはテキスト先頭の一文(あるいは適切に収まる　だけ)を実際の表紙に載せ、続きは隣接したページに載せるべきです。

あなたが文書の「非透過的」複製物を100部を超えて出版あるいは頒布する場合、それぞれの非透過な複製物と一緒に機械で読み取り可能な透過的複製物を添付するか、それぞれの非透過な複製物(あるいはそれに付属する文書)中で、公にアクセス可能なコンピュータネットワーク上の所在地を記述しなければなりません。その場所には、非透過な複製物と内容的に寸分違わず、余計なものが

追加されていない完全な文書の透過的複製物が置かれ、またそこから、ネットワークを利用する一般公衆が、一般に標準的と考えられるネットワークプロ トコルを使ってダウンロードすることができなければなりません。もしあなたが後者の選択肢を選ぶならば、その版の非透過な複製物を公衆に(直接、あるいは　あなたの代理人ないし小売業者が)最後に頒布してから最低1年間は、その透過的複製物が指定の場所でアクセス可能であり続けることを保証するよう、非透 過な複製物の大量頒布を始める際に十分に慎重な手順を踏まなければならなりません。

これは要望であり必要条件ではないが、文書の著者に、文書の更新された版をあなたに提供する機会を与えるため、透過非透過を問わず大量の複製物を再頒布し始める前には彼らにきちんと連絡しておいて下さい。

4. 改変

文書の改変版を、この利用許諾契約書と細部まで同一の契約の下で発表する限り、すなわち原本の役割を改変版で置き換えた形での頒布と改変を、その複製物を所有するすべての人々に許可する限り、あなたは改変版を上記第2項および第3項が指定する条件の下で複製および頒布することができます。さらに、あなたは改変版において以下のことを行わなければなりません。

A. 題扉に(もしあればその他の表紙にも)、文書および文書のそれ以前の版と見分けがつく題名を載せること(もし以前の版があれば、文書の「履歴　(History)」の部分に列記されているはずです)。もし元の版の出版者から許可を得たならば、以前の版と同じ題名を使っても良いです。

B. 題扉に、改変版における改変を行った1人以上の人物　　か団体名を列記すること。あわせて元の『文書』の著者として、最低5人(も し5人以下ならばすべて)の主要著者を列記すること。

C. 改変版の出版者名を出版者として記載すること。

D. 文書にあるすべての著作権表示を残すこと。

E. 他の著作権表示の近くに、あなたの改変に対する適 当な著作権表示を追加すること。

F. 著作権表示のすぐ後に、改変版をこの契約書の条件　の下で利用することを公衆に対して許可する告知を含めること。その形式はこの契約書の末尾にある付記で示されている。

G. 元の『文書』の利用許諾告知に書かれた、変更不可部分の完全な一覧と、要求されるカバーテキストとを、改変版の利用許諾告知でもそのまま残すこと。

H. この契約書の、変更されていない複製物を含めること。

I. 「履歴　(History)」と題された部分とその題名を保　存し、そこに改変版の、少なくとも題名、出版年、新しく変更した部分の著者名、出版者名を、題扉に掲載するのと同じように記載した一項を加えること。もし文書中に「履歴」と題された部分が存在しない場合には、文書の題名、出版年、著者、出版者を題扉に掲載するのと同じように記載した部分を用意し、上記で述べたような、改変版を説明する一項を加えること。

J. 文書中に、文書の透過的複製物への公共的アクセスのために指定されたネットワーク的所在地が記載されていたならば、　それを保存すること。同様に、その『文書』の元になった以前の版で指定されていたネットワーク的所在地も載っていたならば、それも保存すること。これらの情報は「履歴(History)」の部分に置いても良いです。ただし、それが　文書自身より少なくとも4年前に出版された著作物の情報であったり、あるいは改変版が参考にしている版の元々の出版者から許可を得たならば、その情報を削除してもかまいません。

K. 「謝辞 (Acknowledgement)」あるいは「献辞 (Dedication)」等と題されたいかなる部分も、その部分の題名を保存し、その部分の内容(各貢献者への謝意あるいは献呈の意)と語調を保存すること。

L. 『文書』の変更不可部分を、その本文および題名を　変更せずに保存すること。章番号やそれに相当するものは部分の題名の一部とは見做さない。

M. 「推薦の辞 (Endorsement)」というような章名が題された部分はすべて削除すること。そのような部分を改変版に含めてはいけません。

N. すでに存在する部分を「推薦の辞 (Endorsement)」と題されるように改名したり、題名の点で変更不可部分のどれかと抵触するように改名してはいけません。

もし改変版に、補遺部分としての条件を満たし、かつ文書から複製物され た文章や図などをいっさい含んでいない、前書き的な章あるいは付録が新しく含まれるならば、あなたは希望によりそれらの部分の一部あるいはすべてを変更不可と宣言することができます。変更不可を宣言するためには、それらの部分の題名を改変版の利用許諾告知中の変更不可部分一覧に追加すれば良いです。これらの題名は他の章名とは全く別のものでなければなりません。

含まれる内容が、さまざまな集団によるあなたの改変版に対する推薦の辞のみである限り、あなたは、「推薦の辞　(Endorsement)」と題された章を追加することができます。推薦の辞の例としては、ピアレビュー（内容審査）の陳述、あるいは文書がある標準の権威ある定義としてその団体に承認されたという声明などがあります。

あなたは、 5語までの一文を表カバーテキストとして、 25語までの文を裏表紙テキストとして、改変版のカバーテキスト一覧の末尾に加えることができます。 一個人ないし一団体が直接(あるいは団体内で結ばれた協定によって)加えることができるのは、表カバーテキストおよび裏カバーテキストとしてそれぞれ一 文ずつのみである。もし以前すでにその文書において、表裏いずれかの表紙にあなたの(またはあなたが代表する同じ団体内で為された協定に基づく)カバー テキストが含まれていたならば、あなたが新たに追加することはできません。しかしあなたは、その古い文を加えた以前の出版者から明示的な許可を得たならば、古い文を置き換えることができます。

文書の著者あるいは出版者は、この利用許諾契約書によって、彼らの名前を利用することを許可しているわけではありません。彼らの名前を改変版の宣伝に使ったり、改変版への明示的あるいは黙示的な保証のために使うことを許可するものではありません。

5. 文書の結合

あなたは、上記第4項において改変版に関して定義された条件の下で、この利用許諾契約書の下で発表された複数の文書を一つにまとめることができます。その際、原本となる文書にある変更不可部分を全て、改変せずに結合後の著作物中に含め、それらをあなたが統合した著作物の変更不可部分としてその利用許諾告知において列記し、かつ原本にある全ての保証否認警告を保存しなければなりません。

結合後の著作物についてはこの契約書の複製物を一つ含んでいればよく、同一内容の変更不可部分が複数ある場合には一つで代用してよいです。もし同じ題名だが内容の異なる変更不可部分が複数あるならば、そのような部分のそれぞれの題名の最後に、(もし分かっているならば)その部分の原著者あるいは出版者の名前で、あるいは他と重ならないような番号を括弧で括って記載することで、 それぞれ見分けが付くようにしなければなりません。結合後の著作物の利用許諾告知における変更不可部分の一覧においても、章の題名に同様の調整をするこ と。

結合後の著作物においては、あなたはそれぞれの原本の「履歴　(History)」と　題されたあらゆる部分をまとめて、「履歴　(History)」と題された一章にしな　ければなりません。同様に、「謝辞 (Acknowledgements)」あるいは「献辞　(Dedications)」と題されたあらゆる部分もまとめなければならない。あなたは 「推薦の辞 (Endowsements)」と題されたあらゆる部分も削除しなければなりません。

6. 文書の収集

あなたは、この利用許諾契約書の下で発表された複数の文書で構成される収集著作物を作ることがでます。その場合、それぞれの文書が逐語的に忠実に複製されることを保障するために他のすべての点でこの契約書の定める条件に従う限り、さまざまな文書中のこの契約書の個々の複製物を、収集著作物中に複製物を一つ含めることで代用することがでます。

あなたは、このような収集著作物から文書を一つ取り出し、それをこの契約書の下で頒布することができます。ただしその際には、この契約書の複製物を抽出された文書に挿入し、またその他すべての点でこの文書の逐語的に忠実な複製に関してこの契約書が定める条件に従わなければなりません。

7. 独立した著作物の集積

文書あるいはその派生物を、他の別の独立した文書あるいは著作物と一緒 にし、一巻の記憶装置あるいは頒布媒体に収めた編集著作物は、編集に起因する著作権が編集著作物に含まれる個々の著作物がその利用者に許可した法的権利を制限するよう行使されない限り、「集積」著作物と呼ばれます。文書が集積著作物に含まれる場合、この契約書は、文書と共にまとめられた他の独立した著作物には、それら自身が文書の派生物で無い限り適用されることにはなりません。

このような文書の複製物において、この利用許諾契約書の第3項によりカバーテキストの掲載が要求されている場合、文書の量が集積著作物全体の2分の1以下であれば、文書のカバーテキストは集積著作物中で文書そのものの周りを囲む中表紙、あるいは文書が電子的形式である場合には表紙の電子的等価物にのみ配置するだけでよいです。その場合以外は、カバーテキストは集積著作物全体を取り巻く印刷された表紙に掲載されなければなりません。

8. 翻訳

翻訳は改変の一種と見做すので、あなたは文書の翻訳をこの利用許諾契約書の第4項の定める条件の下で頒布することができます。変更不可部分を翻訳によって置き換えるには著作権者の特別許可を必要としますが、元の変更不可部分に追加する形で変更不可部分の全てないし一部の翻訳を含めることはかまいません。 この契約書や文書中の利用許諾告知、保証否認警告すべての英語原本も含める限り、あなたはこの契約書、告知、警告の翻訳を含めることができます。契約書や告知、警告に関して翻訳と英語原本との間に食い違いが生じた場合、英語原本が優先されます。

9. 契約の終了

この利用許諾契約書の下で明確に提示されている場合を除き、あなたは文書を複製、改変、サブライセンス、あるいは頒布してはなりません。このライセンスで指定されている以外の、文書の複製、改変、サブライセンス、頒布に関するすべての企ては無効であり、この契約書によって保証されるあなたの権利を自動的に終結させることとなります。しかし、この契約書の下であなたから複製物ないし諸権利を得た個人や団体に関しては、そういった人々がこの契約書に完全に従ったままである限り、彼らに与えられた許諾は終結しません。

10. 将来における本利用許諾契約書の改訂

フリーソフトウェア財団は、時によってGNU　フリー文書利用許諾契約書の新しい改訂版を出版することができます。そのような新版は現在の版と理念においては似たものになるでしょうが、新たに生じた問題や懸念を解決するため細部に おいては違ったものになるでしょう。詳しく は http://www.gnu.org/copyleft/を 参照して下さい。

GNU　フリー文書利用許諾契約書のそれぞれの版には、新旧の区別が付くようなバージョン番号が振られています。もし文書において、この契約書のある特定の版か「それ以降のどの版でも」適用して良いと指定されている場合、あなたはフリーソフトウェア財団から発行された(草稿として発表されたものを除く)指定の版かそれ以降の版のうちどれか一つを選び、その条項や条件に従うことができます。もし文書がこの契約書のバージョン番号を指定していない場合には、あなたはフリーソフトウェア財団から今までに出版された(草稿として発表されたものを除く)版のうちからどれか一つを選ぶことができます。

. この利用許諾契約書をあなたの文書に適用するには

この利用許諾契約書をあなたが書いた文書に適用するには、この契約書の複製物一つを文書中に含め、以下に示す著作権表示と利用許諾告知を題扉のすぐ後に置いて下さい:

> Copyright （C） 西暦年　あなたの名前.　この文書を、フリーソフトウェア財団発行の　GNU　フリー文書利用許諾契約　書(バージョン1.1かそれ以降から一つを選

> 択)が定める条件の下で複製、頒　布、あるいは改変することを許可します。変更不可部分、表カバーテキスト、　裏カバーテキストは存在しません。この利用許諾契約書の複製物は「GNU フ リー文書利用許諾契約書」という章に含まれています。

もし変更不可部分や表カバーテキスト、裏カバーテキストがあれば、「変更不 可部分...は存在しない。」というところを以下で置き換えてください:

あなたの文書に、他に類を見ない独自のプログラムコードのサンプルが含まれる場合、フリーソフトウェアにおいてそのコードを利用することを許可するために、そういったサンプルに関してはこの利用許諾契約書と同時にGNU　一般公衆許諾契約書のようなフリーソフトウェア向けライセンスのうちどれか一つを選択して適用してもよい、というような条件の下で発表することを推奨します。