

Documentation

OTRS Business Solution™ 6 Manual

Build Date:

2017-09-27

OTRS Business Solution™ 6 Manual

Copyright © 2017 OTRS AG

This work is copyrighted by OTRS AG.

You may copy it in whole or in part as long as the copies retain this copyright statement.

All trade names are used without the guarantee for their free use and are possibly registered trade marks. All products mentioned in this manual may be trade marks of the respective manufacturer.

The source code of this document can be found at [github](#), in the repository [doc-otrsbusiness](#). Contributions are more than welcome. You can also help translating it to your language at [Transifex](#).



Table of Contents

1. Introduction	1
2. Upgrading to OTRS Business Solution ™	2
3. Features of OTRS Business Solution ™	3
1. The Chat	3
1.1. Description	3
1.2. Configuration	3
1.3. Setup	5
1.4. General Usage	6
1.5. Advanced Usage	11
2. The Dynamic Field <i>Contact with Data</i>	14
2.1. Description	15
2.2. Configuration	15
2.3. Usage	16
3. The Dynamic Field <i>Database</i>	16
3.1. Description	17
3.2. Configuration	17
3.3. Usage	21
4. The Dynamic Field <i>Web Service</i>	22
4.1. Description	23
4.2. Configuration	23
4.3. Usage	28
5. Import Ready2Adopt Process	28
5.1. Import	28
6. Import Ready2Adopt Web Services	29
6.1. Import	29
7. SMS Notifications and Notification Web View	30
7.1. Description	30
7.2. Setting up SMS Notifications	31
7.3. Setting up and Using the Notification Web View	32
8. The Report Generator	33
8.1. Description	33
8.2. Configuration	33
8.3. Usage	33
9. Resource Planning	37
9.1. Description	37
9.2. Configuration	37
9.3. Usage	37
10. The SLA Field Selection Dialog	42
10.1. Description	42
10.2. Configuration	42
10.3. Usage	43
11. SMS Communication Channel	43
11.1. Description	43
11.2. Activating SMS Cloud Service	43
11.3. Configuring SMS Templates	45
11.4. Creating SMS Tickets	45
11.5. Creating SMS Articles	46
11.6. Displaying SMS Articles	47
12. The Article Attachment Overview	48
12.1. Description	48
12.2. Configuration	48
12.3. Usage	48
13. The Ticket Timeline View	49
13.1. Description	49
13.2. Configuration	49


13.3. Usage	49
14. System Configuration User Specific Settings	50
14.1. Description	50
14.2. Configuration	50
14.3. Usage	50
15. System Configuration History	52
15.1. Description	52
15.2. Configuration	52
15.3. Usage	52
A. Additional Resources	54



List of Figures

2.1. Upgrading Screen	2
2.2. Upgrade Successful	2
3.1. Setting up preferred chat channels	5
3.2. Setting up Chat Availability	6
3.3. The Chat Managing Screen	6
3.4. Starting an A2C chat from ticket zoom	7
3.5. Starting an A2C chat from dashboard	7
3.6. Starting a C2A chat	7
3.7. An Open Chat Request	8
3.8. Open Chat Request Details	8
3.9. The chat widget	9
3.10. Monitor a chat	9
3.11. Advanced chat actions	9
3.12. The chat widget	10
3.13. Configure Chat Integration Button	11
3.14. Configuration Tab	12
3.15. Integration Code Tab	13
3.16. Preview Tab	14
3.17. Dynamic Field Database Admin screen	18
3.18. DynamicField Database - Historical data settings	20
3.19. DynamicField Database - Test field	21
3.20. DynamicField Database - Autocomplete feature	21
3.21. DynamicField Database - Selected items	21
3.22. DynamicField Database - Detailed search	22
3.23. DynamicField Database - Detailed search result	22
3.24. Dynamic Field <i>Web Service</i> - Outgoing mapping example	24
3.25. Dynamic Field <i>Web Service</i> - Incoming mapping example	24
3.26. Dynamic Field <i>Web Service</i> - Definition Example	27
3.27. Import Ready2Adopt Process Widget	29
3.28. Add web service	30
3.29. Import Ready2Adopt Web Service	30
3.30. Enabling SMS Notifications	31
3.31. Agent Preferences SMS Notification Settings	31
3.32. SMS Notification Text Fields	32
3.33. Enabling the Notification Web View	32
3.34. Notification Web View Icon on the Toolbar	32
3.35. Notification Web View Overview	32
3.36. Notification Web View Detail	33
3.37. Automatic generation settings	33
3.38. Automatic generation settings	34
3.39. Output settings	34
3.40. Report: Front page	35
3.41. Report: Table of contents	35
3.42. Report: Preamble	36
3.43. Report: Statistics	36
3.44. Team Management Add and Import	37
3.45. Team Management Add and Import	38
3.46. Team Agent Management	38
3.47. Team Agent Relations	38
3.48. Assigning team and agent to an appointment	39
3.49. Assigning multiple teams and agents to an appointment	40
3.50. Resource Overview	40
3.51. Resource Overview	41
3.52. Resource Overview Settings	41
3.53. Resource Overview Filter Delete	42

3.54. SLA configuration	43
3.55. SLA message	43
3.56. Cloud Services Screen in Admin Interface	43
3.57. Activating SMS Cloud Service	44
3.58. Manage SMS Templates	45
3.59. Manage SMS Templates	45
3.60. New SMS Ticket Screen	46
3.61. New SMS Ticket Screen	46
3.62. New SMS Ticket Screen	47
3.63. SMS Article Display	47
3.64. SMS Ticket History Entries	48
3.65. Switching to the article attachment overview	48
3.66. Using the article attachment overview	48
3.67. Switching to the Ticket Timeline View	49
3.68. Using the Ticket Timeline View	50
3.69. Accessing the user specific settings	51
3.70. Reverting a setting to default value	51
3.71. Access to user values of a particular setting	51
3.72. Usage of the user specific values screen	51
3.73. Access to the setting reset dialog	52
3.74. Choosing the reset scope	52
3.75. System configuration deployment history	53
3.76. System configuration setting history	53

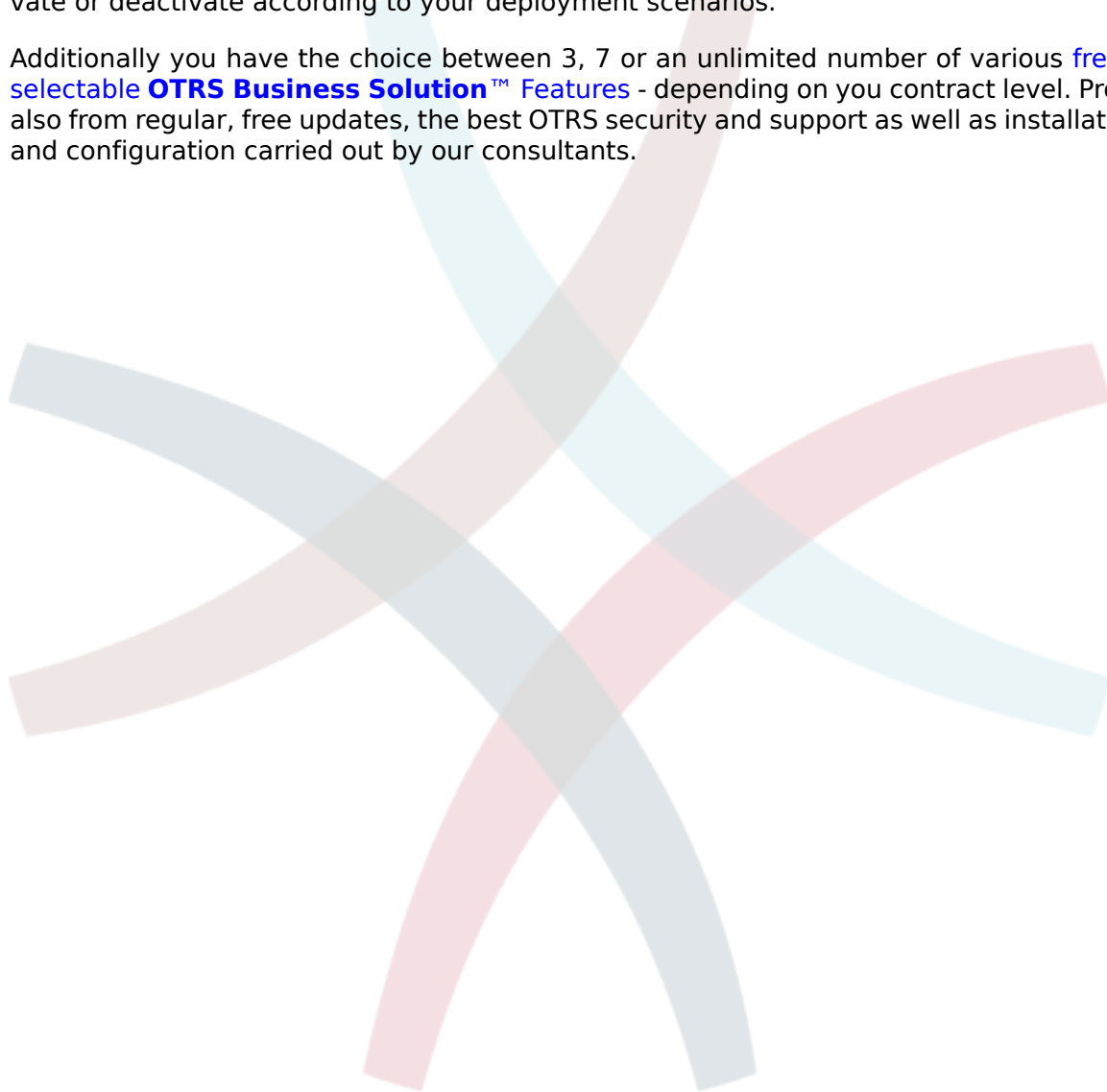


Chapter 1. Introduction

With the **OTRS Business Solution™** you can enjoy all benefits of professional services and exclusive Business Features offered by the OTRS Group. Optimize your internal & external communication for more service quality – flexibly adjustable to your operating scenario.

The sleek and fast framework of OTRS 6 is the base for the **OTRS Business Solution™**. On top are available for you additionally an exclusive set of features. These include the **integrated Cloud Services** of the **OTRS Business Solution™** that you can flexibly activate or deactivate according to your deployment scenarios.

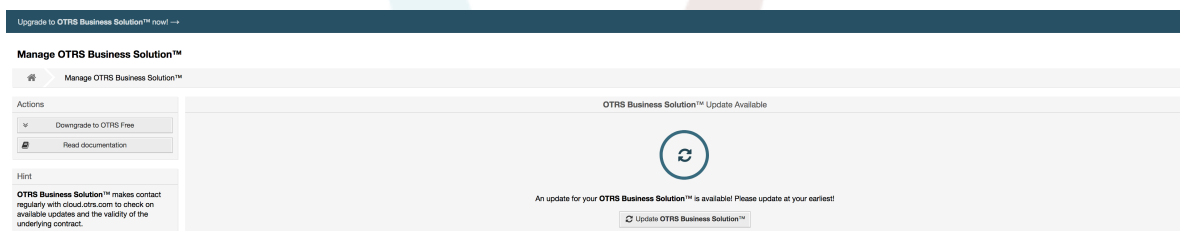
Additionally you have the choice between 3, 7 or an unlimited number of various **freely selectable OTRS Business Solution™ Features** - depending on you contract level. Profit also from regular, free updates, the best OTRS security and support as well as installation and configuration carried out by our consultants.



Chapter 2. Upgrading to OTRS Business Solution™

In order to upgrade to **OTRS Business Solution™**, you need a valid contract with the OTRS group. If you already have a contract, you can upgrade to **OTRS Business Solution™** by visiting the **OTRS Business Solution™** administration screen in your OTRS instance. You can find the link to this screen in the *System Administration* section of the admin area of your OTRS Free instance. Please note that you first have to register your system using the system registration mechanism.

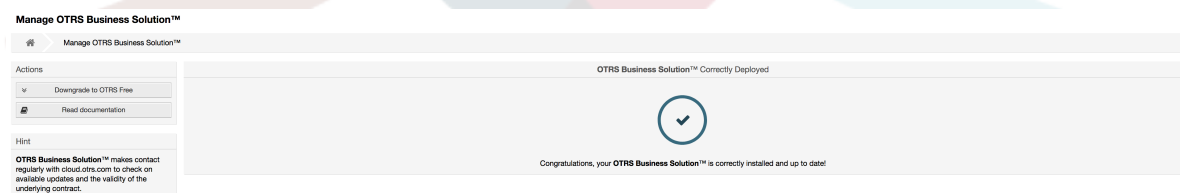
Figure 2.1. Upgrading Screen



If you're entitled to upgrade, you will (as an admin user) also see a notification bar which guides you to the upgrade screen. The upgrade itself is done automatically by just pressing the upgrade button. Please note that you don't need to confirm the upgrade again. Pressing the upgrade button immediately starts the upgrading process.

If the upgrade was successful, you will see a confirmation on the **OTRS Business Solution™** administration screen.

Figure 2.2. Upgrade Successful



If you have any questions regarding your entitlement for the **OTRS Business Solution™** or technical questions, please contact sales@otrs.com.

Chapter 3. Features of OTRS Business Solution™

1. The Chat

1.1. Description

The chat feature of **OTRS Business Solution™** allows for chats between agents and customers using the OTRS frontend. Once enabled, customers (or public users) can start chats with agents and vice versa. Finished chats can be archived and attached to tickets, or can be downloaded as PDF by agents, customers and public users.

Each chat is assigned to a chat channel. Chat channels are like chat rooms. Each chat channel is assigned to a certain group which controls the permissions for this channel. If no chat channel is selected when a chat is created, this chat will be created in a default channel (which is automatically created if it doesn't exist).

1.2. Configuration

The following system configuration options are relevant for this feature. Please note that you can also define a list of fixed texts which are being displayed to public users and customers in the chat module. For more information on which texts can be changed, please have a look at the module configuration (OTRSBusiness → Core::Chat).

`ChatEngine::Active`

Controls whether or not the chat feature is enabled.

`ChatEngine::ChatDirection::PublicToAgent`

Controls whether or not public users should be able to start chats (please note that the frontend module configuration for the public chat frontend in `PublicFrontend::Module###PublicChat` has to be enabled, too).

`ChatEngine::ChatDirection::CustomerToAgent`

Controls whether or not customers should be able to start chats (please note that the frontend module configuration for the customer chat frontend in `CustomerFrontend::Module###CustomerChat` has to be enabled, too).

`ChatEngine::ChatDirection::AgentToCustomer`

Controls whether or not agents should be able to start chats with customers directly.

`ChatEngine::ChatDirection::AgentToAgent`

Controls whether or not agents should be able chat with each other.

`ChatEngine::PermissionGroup::ChatReceivingAgents`, `ChatEngine::PermissionGroup::ChatNotificationAgents`, `ChatEngine::PermissionGroup::ChatStartingAgents`

Defines groups for the certain features. Receiving: Groups which can receive and accept chat requests. Notification: Groups which can receive notifications about new/pending chat requests. Starting: Groups which are allowed to start new chats at all.

ChatEngine::DefaultAgentName

A default name which should be displayed to customers and public users instead of the real name of the chatting agents. If empty or disabled, the real agent name will be shown.

ChatEngine::DefaultAgentNameNumbers

Controls whether or not numbers should be added to the agents default name in case more than one agent is in a chat to allow customers/public users to distinguish between the different agents (e.g. "Support Agent 1", "Support Agent 2" etc.).

ChatEngine::PublicInterface::AllowChatChannels

Defines, if chat channel selection should be enabled for public users. If set to "no", chats from public users will go to the default channel automatically (ChatEngine::DefaultChatChannel).

ChatEngine::PublicInterface::AvailabilityCheck

Defines, if chat channels should only be selectable in the public interface if at least one agent with sufficient permissions is available in this channel.

ChatEngine::CustomerInterface::AllowChatChannels

Defines, if chat channel selection should be enabled for customer users. If set to "no", chats from customer users will go to the default channel automatically (ChatEngine::DefaultChatChannel).

ChatEngine::CustomerInterface::AvailabilityCheck

Defines, if chat channels should only be selectable in the customer interface if at least one agent with sufficient permissions is available in this channel.

ChatEngine::CustomerThreshold

Amount of minutes which need to pass before a *no agent is available* message is being displayed to the customer/public user who started the chat request. The message can be configured using ChatEngine::Texts::CustomerFrontend::NoAgentsAvailable::NewTicket or ChatEngine::Texts::CustomerFrontend::NoAgentsAvailable::AddToExisting (customer interface) and ChatEngine::Texts::PublicFrontend::NoAgentsAvailable (public interface).

ChatEngine::AgentOnlineThreshold

If an agent is not active in OTRS within this threshold, they will be automatically set to *unavailable for chat* (see Agents: Set up availability).

ChatEngine::DefaultChatChannel

Name of the default chat channel. This channel must not be created manually and can't be deleted. This channel is taken as target chat channel in case chat channel selection is disabled for any interface.

ChatEngine::ChatTTL

After how many hours a closed chat should be deleted from the database. Closed chats can still be viewed and downloaded by customers. After a chat has been deleted from the database, it will no longer be possible for customers to view or download it.

ChatEngine::ChatDecayTime

After how many days chats which are not closed should be deleted from the database automatically in order to keep the database clean. Please note that this setting can

possibly delete chats which are still in use if they are older than the configured amount of days.

ChatEngine::ChatOrder

Defines, if new chats should be prepended or appended to the list of active chats in the active chats widget in the chat manager.

1.3. Setup

In order to be able to create your first chat, you need to setup certain things in your OTRS instance.

1.3.1. Admin: Create chat channels and assign permissions

In the admin area in OTRS, you'll find a new entry *Chat Channel* (AdminChatChannel). Use this screen to add new channels. You'll need to assign an existing agent group to each channel you'll create. You can use the screens *Agents ↔ Groups* or *Agents ↔ Roles* to assign permissions. There are new chat-specific permissions which are described below.

Available Chat Permissions

CHAT_OBSERVER

Users with this permission type will only be able to observe chats in this channel after they have been invited to them. They will not be able to accept or observe chats in this channel on their own and if they are the only agents currently available in this channel, customers/public users will not be able to create a new chat in this channel. Observers can still invite other agents to a chat they currently observe.

CHAT_PARTICIPANT

Users with this permission type will be able to take part in a chat, but only after they get invited to it. After they're in a chat, they can change the chat channel, discard the chat etc. They will not be able to accept a customer/public user chat request on their own.

CHAT_OWNER

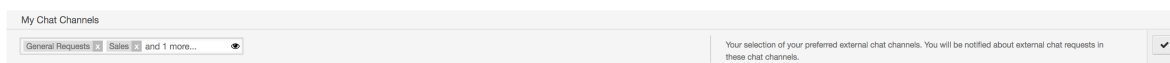
Users with this permission type will be able to accept chat customer/public requests and do all kinds of observer and participant actions on a chat.

Note

To use the chat in general, agents still need to have at least ro permissions to the groups set up in ChatEngine::PermissionGroup::ChatReceivingAgents and ChatEngine::PermissionGroup::ChatStartingAgents.

1.3.2. Agents: Select Active Chat Channels

Figure 3.1. Setting up preferred chat channels

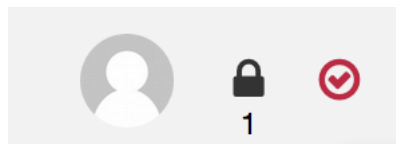


As an agent, you'll need to select in which chat channels you want to be available. Customers will only be able to create a new chat request on a certain channel if at least one

agent with owner permissions has selected this channel in their preferences and is set as available for external chats (see next section).

1.3.3. Agents: Set up Availability

Figure 3.2. Setting up Chat Availability



To be available for any chats, you'll need to use the chat toolbar switch. It's a three-way switch with these states: unavailable for chats (empty circle), available for internal (agent-to-agent) chats (empty circle with tick sign), available for internal and external (customer-to-agent, public-to-agent) chats (filled circle with tick sign). Just click the toggle in order to change your availability.

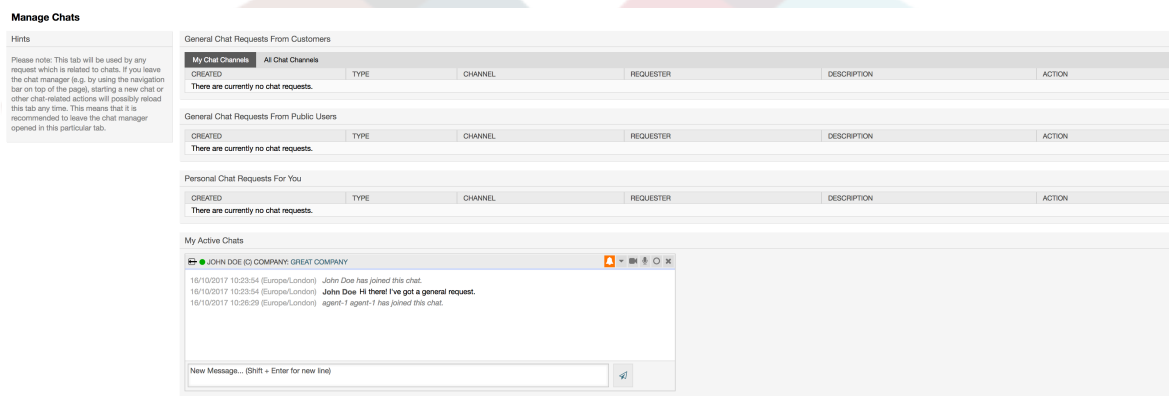
Note

Your availability will be reset to *internal chats only* each time you log in to OTRS again.

1.4. General Usage

1.4.1. The Chat Manager

Figure 3.3. The Chat Managing Screen



The chat manager is the central point of the chat feature. As an agent, here you can manage your chats and see and accept open requests. The chat manager provides a list of chat requests from customers, public users and other agents, as well as an overview over all active chats. Open the chat manager by using *Chat* in the main navigation of OTRS.

1.4.2. Starting New Chats

- *Agent to agent*

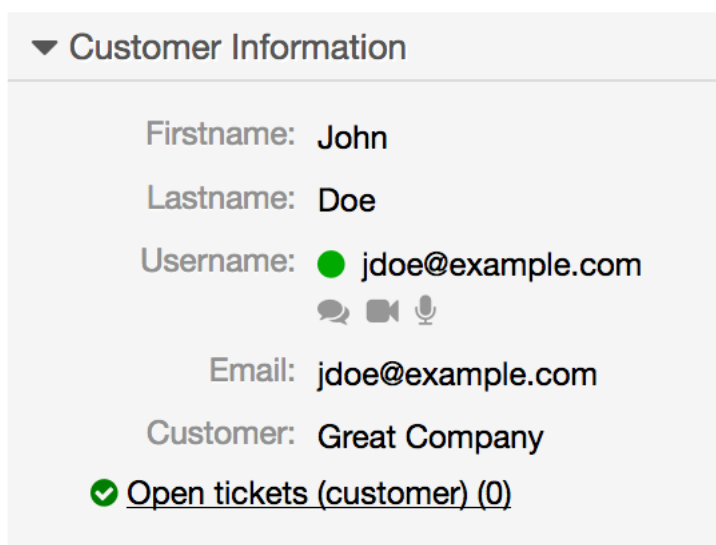
To start a chat with another agent, you can use the user online widget from the dashboard. Agents who are able to use the chat, will have a chat icon next to their name which you can click on. Once clicked, a chat form will open which you can use to enter your first chat message to this agent. After the chat request has been sent to the other agent, you can watch the chat in the chat manager.

- *Agent to customer*

To start a chat with a customer, you can use the customer user widget from the customer information center when viewing a certain customer. Online customers will have a chat icon next to their name which you can click on. Once clicked, a chat form will open which you can use to enter your first chat message to this customer user. After the chat request has been sent to the customer, you can watch the chat in the chat manager.

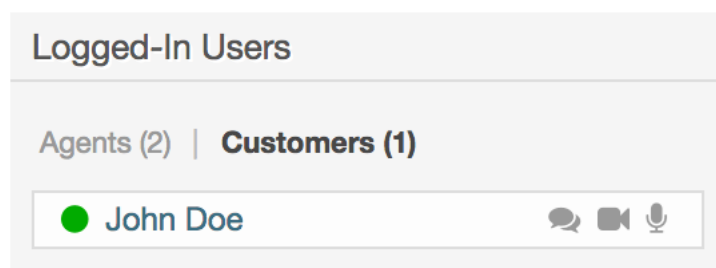
Another possibility is to create a chat from the ticket zoom. If the customer user who is selected in this ticket is currently online, you'll be able to use a chat icon next to their username in the customer information box in right sidebar (see figure below).

Figure 3.4. Starting an A2C chat from ticket zoom



Additionally, you will be able to start chat with customer users by clicking on chat icons next to their name in the user online widget on the dashboard.

Figure 3.5. Starting an A2C chat from dashboard



- *Customer to agent*

Customers are not able to start a chat directly with a certain agent, but only to create generic chat requests. To do this, customers can use the item *Chats* → *Create new chat* from the customer interface main navigation.

Figure 3.6. Starting a C2A chat



- *Public to agent*

Users of the public interface are also not able to start a chat directly with a certain agent, but only to create generic chat requests using the public chat module (`public.pl?Action=PublicChat`). If you would like to integrate the public chat module into your website, you can use the `IsIframe` parameter. The OTRS header and footer will then be removed from the view (`public.pl?Action=PublicChat;IsIframe=1`).

1.4.3. Handling Chat Requests

If there is a new chat request in one of the channels you (as an agent) have permissions for, you'll receive a browser notification which allows you to go directly to the chat manager. After you've opened the chat manager, you can see a list of open requests of all types (see figure below).

Figure 3.7. An Open Chat Request

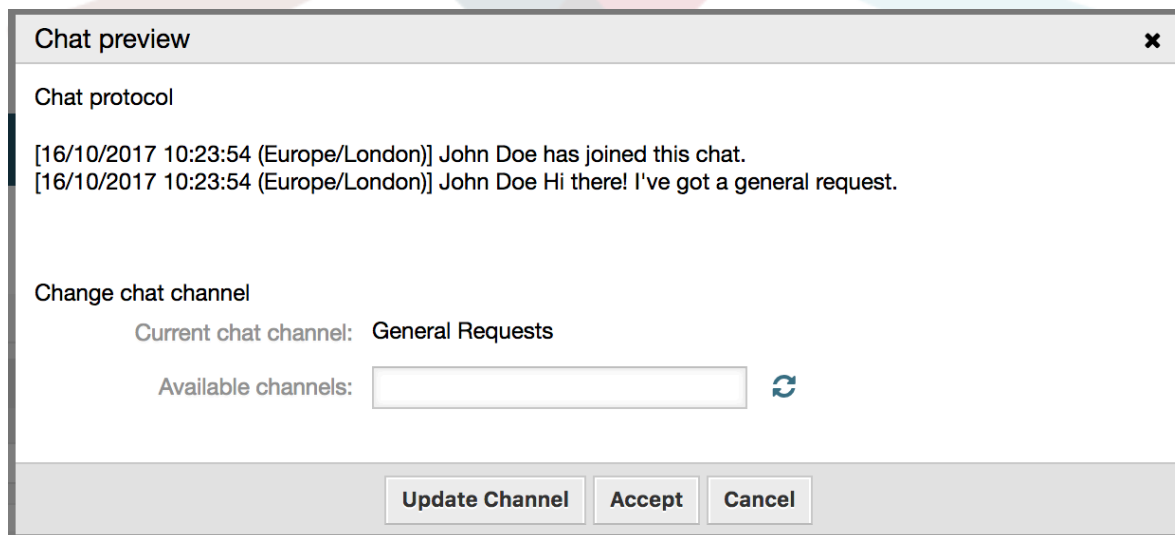


CREATED	TYPE	CHANNEL	REQUESTER	DESCRIPTION	ACTION
16/10/2017 10:23:54 (Europe/London)	Customer	General Requests	John Doe	Hi there! I've got a general request.	Open chat

Use the *Open chat* button from the list to see the details of the request. Clicking the button will open an overlay which provides several actions on this request. Given that you've got sufficient permissions, you can accept the request or change the channel for this request. Changing the channel will follow the same rules as creating new request: you can only move the request to a channel which has active agents in it.

Changing the chat channel without accepting the request could be used to dispatch chats to the right department for example.

Figure 3.8. Open Chat Request Details



Chat preview ✕

Chat protocol

[16/10/2017 10:23:54 (Europe/London)] John Doe has joined this chat.
 [16/10/2017 10:23:54 (Europe/London)] John Doe Hi there! I've got a general request.

Change chat channel

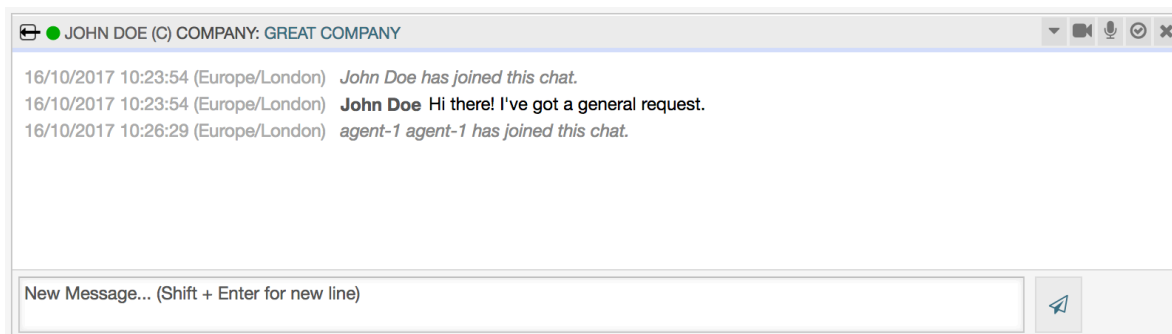
Current chat channel: General Requests

Available channels: ↻

If you want to enter a chat, you can use the *Accept* button from the chat detail overlay box. The chat will then be removed from the list of requests, added to the list of your active chats and the customer/public user will receive a message that you've entered the chat.

1.4.4. The Chat Widget

Figure 3.9. The chat widget



The chat widget is the main component you'll use when chatting with other people. It provides a history of all messages in the chat as well as list of possible actions (depending on your permission level).

To send messages, you can use the text field in the chat widget in your list of active chats. Once you are done with typing your message, submit it by either clicking the send icon next to the text field, or using tab and enter (consecutively). When the message has been sent, the cursor will jump back to the text field to allow for easily continuing to type.

Following you'll find a list of possible actions in the chat widget.

- *Action: Monitor & close*

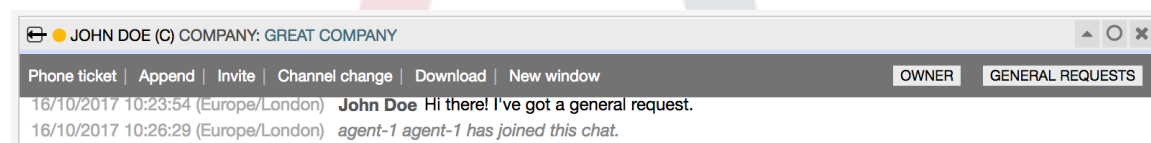
From the chat widget head top right hand side, you can close the chat by clicking the "x" icon. Clicking the circle icon right next to the close icon, allows you to set up a personal monitoring for this chat, which is again a three-way switch (no monitoring = empty circle, only monitor customer activity = empty circle with tick icon, monitor all actions = filled circle with tick icon). When monitoring a chat, browser notifications will show up each time there is a new action in this chat.

Also from the header, you can expand a toolbar of advanced chat actions for this chat using the triangle icon. The toolbar also shows you the channel this chat takes place in and which role you have in this chat (e.g. Owner).

Figure 3.10. Monitor a chat



Figure 3.11. Advanced chat actions



- *Action: Phone ticket*

Using the *Phone ticket* link from the advanced toolbar, allows you to create a phone ticket to which the current chat will be appended to as an article. After you've successfully created the ticket, the chat will get closed automatically.

- *Action: Append*

Allows you to append this chat as an article to a selected ticket. After the article has been created, the chat will get closed automatically.

- *Action: Invite*

Allows you to invite another agent to this chat. You can select from a list of available agents who you want to invite. Once you've invited the agent, a new agent-to-agent chat request will be sent to this agent. After the request has accepted, a new internal chat will be established between you and the invited agent. In this internal chat, the invited agent has two more actions available in the advanced actions toolbar: Observer and Participant.

- *Action: Channel change*

Using this link, you can change the channel for this chat. For the selection of the target channel the same rules apply as for initially selecting a chat channel.

- *Action: Download*

Lets you download the complete history of this chat as PDF.

- *Action: New window*

Opens this chat in a separate popup window. You can use this window in the same way as the chat widget and both the widget and the popup can be used at the same time.

- *Action: Observer*

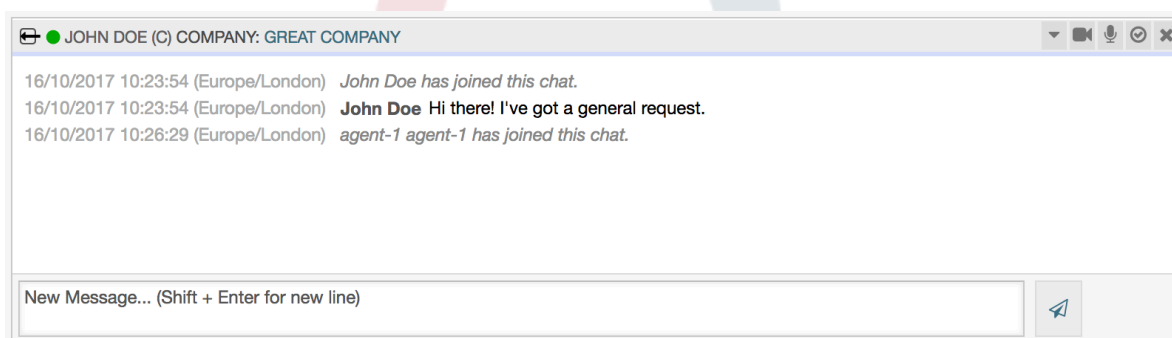
When you have been invited to a chat by another agent, you can decide if you want to join the chat as observer or participant (depending on your permission level). Once you click the action from your toolbar, a new chat widget will be added to the list of your active chats which is the chat you have been invited to. If you join the chat as observer, you will only be able to read what others are contributing to the chat and you will be invisible to customers or public users. Agents will still receive a message that you joined the chat as observer. As an observer, you have the possibility to become a participant anytime by using the advanced action toolbar in your chat widget (depending on your permission level).

- *Action: Participant*

If you want to contribute something to the chat you have been invited to, you can join it as a participant. Customers/public users and agents will receive a message that you have entered the chat. As a participant, you can become an observer anytime by using the advanced action toolbar in your chat widget.

1.4.5. User availability

Figure 3.12. The chat widget



Every chat participant has colored icon (circle) before his/her name in the *Chat widget* header, which represents participant availability. Icon color represents participant state as follows:

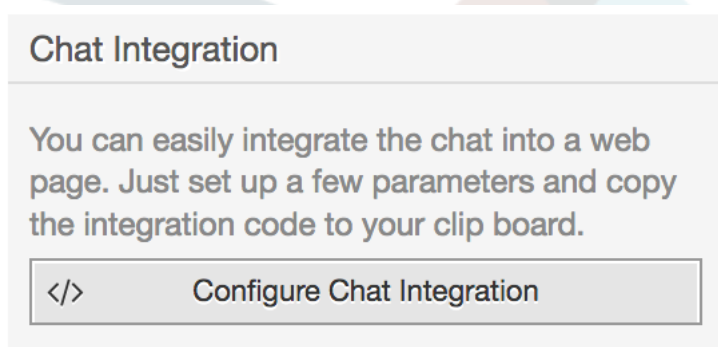
- *Active* - participant is active (available and online).
- *Away* - participant is away (available, but there was no request sent to the server from this user for some time - default 5 minutes, see `ChatEngine::AgentOnlineThreshold`).
- *Inactive* - participant is inactive (chat request is not accepted yet or participant left the chat).
- *Unavailable* - participant is unavailable (set via *Availability* button).
- *Offline* - participant has logout or session has expired.

1.5. Advanced Usage

1.5.1. Chat Integration in Web Pages

You can easily integrate the public chat into a website. In the chat manager, administrators can access a button to open public chat integration configuration and snippet generator.

Figure 3.13. Configure Chat Integration Button



1.5.1.1. Configuration

Configuration tab contains several parameters that can be tweaked to suit your needs. While most of them are self-explanatory, *General Settings* section contains several important options:

Display Chat if no agent is available

If enabled, this option controls whether the chat will be displayed regardless of agent's presence in the system. If not enabled, if there are no agents currently online and available for chat, widget will be hidden.

Fall-back URL

This option is taken into account only if above option is enabled. If there are no agents present and after some time, users of the public chat will be presented with a message containing this link.

Pre-selected channel

Default chat channel selection for public chat requests. This chat channel will be pre-selected in the chat widget, but users will be able to change it if needed.

Figure 3.14. Configuration Tab

Chat Integration
✕

You can easily integrate the public chat into a website. Just copy the code snippets from here. If you would like to make any adaptations, please switch to the configuration tab. All changes are applied immediately to the snippet, so you can make your changes and copy the snippet to your clipboard afterwards. Please note that changes made on this page are not remembered.

Configuration
Integration Code
Preview

General Settings

Display Chat if no agent is available

Fall-back URL (only applies if above is set to "Yes")

Pre-selected channel

Layout & Colors

Position of the chat widget

Delay (in milliseconds) until the chat shows up

Height in pixels to which the chat should expand

Should the chat already be maximized when it shows up?

Background color of the inner chat window

Background color of the headline on top of the chat

Text color of the headline on top of the chat

Background color of the send button

Text color of the send button

Text color of the chatter Name in the chat

Texts

Headline

Placeholder for the channel selection

Placeholder for the name input

Placeholder for the message input

Confirmation message if the user decides to close the chat

Message which is displayed after the chat was ended by an agent

All other options pertains layout, colors and texts used by the chat widget. You can tweak them to match your overall website theme.

Note

Please note that changes made in the configuration tab will not be saved. Every time you open this dialog, all fields will be reset to default values.

1.5.1.2. Integration Code

Integration Code tab contains necessary code snippets for insertion in a website, so public chat can work. All changes in the *Configuration* tab are applied immediately to the snippets, so you can make your changes and copy the code to your clipboard right afterwards.

Figure 3.15. Integration Code Tab

Chat Integration ✕

You can easily integrate the public chat into a website. Just copy the code snippets from here. If you would like to make any adaptations, please switch to the configuration tab. All changes are applied immediately to the snippet, so you can make your changes and copy the snippet to your clip board afterwards. Please note that changes made on this page are not remembered.

Configuration
Integration Code
Preview

Integration Code

Integrate this into the <head> section of your page:

Integrate this into the bottom of your page (after jQuery has been loaded):

```

<link rel="stylesheet" href="https://otrs-devel/otrs6-ob-web/skins/Custom/default/css/Core.Pt
<script type="text/javascript" src="https://otrs-devel/otrs6-ob-web/js/c
<script type="text/javascript">
Core.Public.Chat.Init({
  OTRSURL : "https://otrs-devel/otrs6-ob/",
  ShowIfNoAgentAvailable : "0",
  iFrameURLFallback : "",
  PreselectedChannelID : "0",
  Position : "right",
  FadeInDelay : "1000",
  ExpandHeight : "36px",
  ExpandToFullHeight : "0",
  BackgroundColor : "#fff",
  HeadlineBackground : "#999",
  HeadlineTextColor : "#fff",
  SendButtonBackground : "#333",
  SendButtonTextColor : "#fff",
  ChatterNameColor : "#f92",
  Headline : "How can we help you?",
  PlaceholderChannel : "Please select a topic for your request...",
  PlaceholderName : "Your name...",
  PlaceholderText : "Your message...",
  CloseConfirmMessage : "Are you sure you would like to end the chat?",
  ChatClosedMessage : "Thank you! The chat was ended."
});
</script>

```

Close

Code from the first field should be added to the <HEAD> section of your web page.

Code from the second field should be inserted into the bottom of the page: before closure of the <BODY> tag, but after the line that loads *jQuery* library.

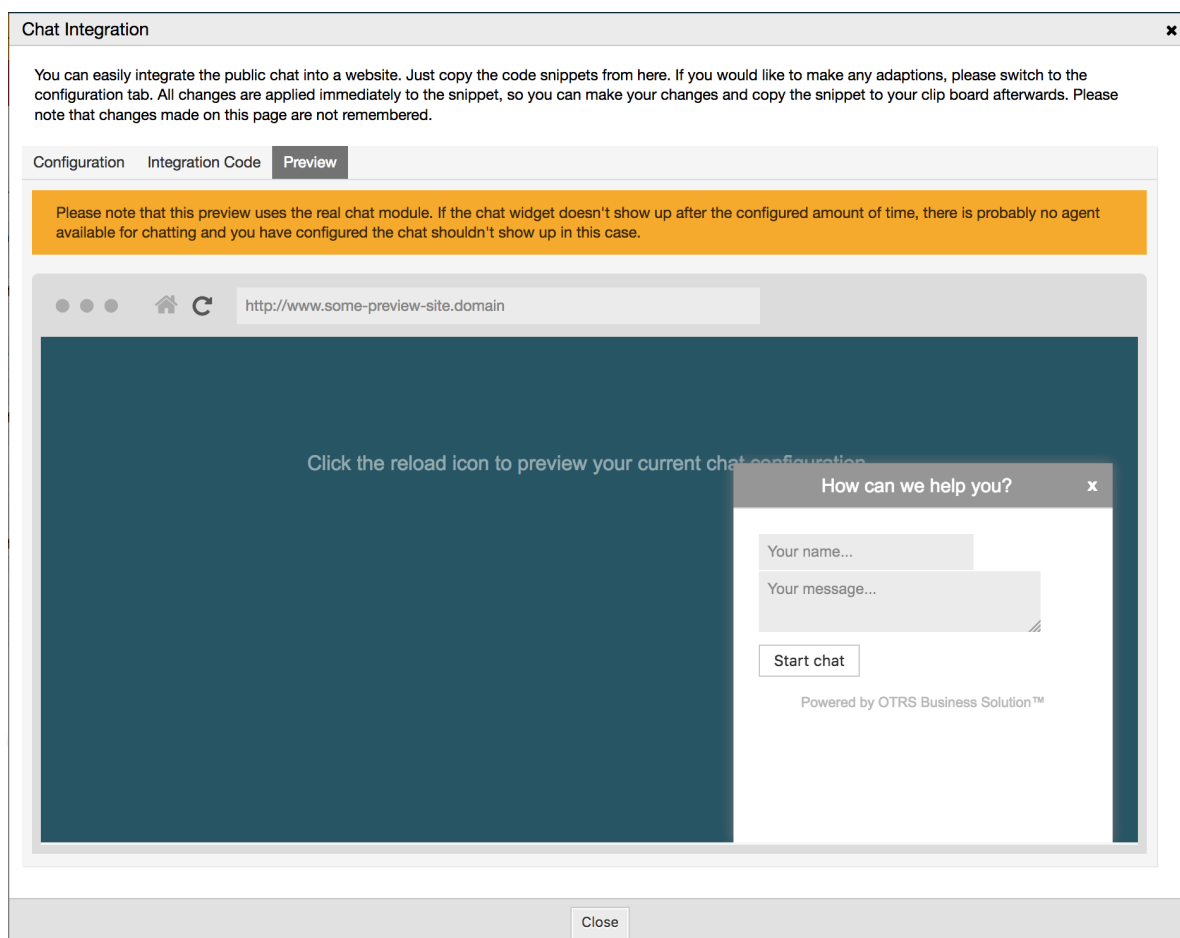
Note

Please note that your web page must also include *jQuery* library for the code snippets to work. This can be recognized by reference to the `jquery.js` or `jquery.min.js` file somewhere in the page.

1.5.1.3. Preview

Using the *Preview* tab you can see the chat widget in action. In the browser simulator below, you can see the chat widget look based on current configuration.

Figure 3.16. Preview Tab



By clicking on the reload icon, you can refresh the simulation at any time.

Note

Please note that the preview uses the real chat module. If the chat widget doesn't show up after the configured amount of time, there are probably no agents available for chatting and you have configured the chat shouldn't show up in this case.

1.5.1.4. Troubleshooting

- If you cannot see the chat widget, but everything seems in order, make sure you have at least one agent who is externally available for chat. Or switch on the *Display Chat if no agent is available* option in the *Configuration* tab, and refresh the code.
- If you get mixed-content warnings in your browser console, make sure that OTRS configuration option `HttpType` is properly set. For example, if you are running OTRS on SSL, this configuration option must be set to `https`. Your website must run on the same protocol for chat widget to work.
- If you get errors saying that `$` variable is undefined, make sure that *jQuery* library is included in your web page, and that chat related code is placed below it.

2. The Dynamic Field *Contact with Data*

This feature allows to add contacts with data to tickets.

2.1. Description

2.1.1. Definition of Data Sources

The data sources for this feature reside on the Dynamic Fields of the new *Contact with data* type, this mean that each dynamic field of this type is a new data source. The name of the dynamic field will become the name of the data source, as they are both the same.

For each data source (dynamic field) a list of contact attributes can be defined (each attribute can contain only one text based value). The contact attributes can be set as mandatory and searchable if needed. The sort order of the attributes can be also defined.

Contact attributes for each data source could contain any information about the contact as needed, such as different addresses, telephone, birthday, anniversaries, favorite food, hobbies, etc.

Note

The attributes Name and ValidID are always mandatory and they are not automatically added, so for each new data source these attributes must be added manually.

Within the data source definition (or dynamic field configuration) they must be represented by the keys Name and ValidID respectively while the values could be *Name* and *Validity* for example.

2.1.2. Contact Management

After having at least one data source (Contact with data Dynamic Field) defined, contacts information can be added to the source to populate it.

Apart from the Name and ValidID, each data source could contain different contact information as defined by the data source (e.g. a data source could have an E-mail field while another could contain Telephone and/or Mobile fields).

The management (adding or updating) contact for the data sources is done by an special screen that can be called from the Ticket menu (*Edit contacts with data*) in the main navigation bar.

2.1.3. Adding Contacts to Tickets

For each configured contact with data dynamic field, a contact created specifically for this dynamic field (or data source) can be chosen from any ticket create or ticket action screen where the dynamic filed has been previously configured.

The contact attributes will be shown in the ticket zoom screen.

2.1.4. Usage of Contacts in Search and Statistics

Tickets can be found in search and statistics via the required attribute Name.

The Name can also be returned as a field value.

2.2. Configuration

System Configuration

AdminContactWithData::RunInitialWildcardSearch

Group: OTRSBusiness, Subgroup: Frontend::Admin::AdminContactWithData

Defines if a search with placeholder should be executed when initially calling the contact management.

Frontend::Module###AdminContactWithData

Group: OTRSBusiness, Subgroup: Frontend::Agent::ModuleRegistration

Allows to distribute contact and contact detail management permissions via group definition.

2.3. Usage

An exemplary usage of contacts with data is as follows:

1. Create a new dynamic field of type *Contact with data*.
2. Set the possible contact attributes (possible values). Name and ValidID are required for any Contact with data Dynamic Field.
 - Add Name attribute (Key: Name, Value: *Name*).
 - Add ValidID attribute (Key: ValidID, Value: *Validity*).
 - Add any other attribute such as Telephone attribute (Key: Telephone, Value: *Phone*).
3. Add the list of mandatory attribute keys comma separated (Name and ValidID are not needed).
4. Set the attribute key order list comma separated as: Name,Telephone,ValidID.
5. Add the list of searchable attribute keys comma separated (Name is not needed).
6. Populate the data source by adding at least one contact in the newly created data source by using *Tickets* → *Edit contacts with data* screen from the main navigation bar.
7. Add the new dynamic field to the screen's configuration where it should be shown. For example in *New Phone Ticket* screen by updating the system configuration setting: `Ticket::Frontend::AgentTicketPhone###DynamicField` and do the same for `Ticket::Frontend::AgentTicketZoom###DynamicField`.
8. Go to *New phone ticket* screen, and notice that the new field is there. Add all ticket needed information.
9. Select an existing contact using autocomplete and choosing a contact.
- 10The assigned contact and its attributes will be shown in the ticket zoom screen.
- 11It is possible to update the attributes of the contact by clicking the *Edit contact data* button that appears in the right side of the title of the contact data box (if the current user is a member of the groups defined in system configuration setting `Frontend::Module###AdminDynamicFieldContactWithData`).
- 12If is necessary to change the contact for this ticket, it can be done via any other ticket action where the dynamic field is configured for display.

3. The Dynamic Field *Database*

This feature implements a dynamic field of the type *Database*.

3.1. Description

This feature implements a generic dynamic field type, which offers the possibility to gather data from external databases. Such connected datasets can be searched and filtered using additional masks.

Related found and marked datasets can be saved to the particular tickets through the dynamic field.

Database dynamic fields can be created the same way, as the default dynamic fields are created.

3.2. Configuration

The following list of configuration options can be used for this feature.

3.2.1. Adding a *Database* Dynamic Field

This feature implements a configuration interface to create dynamic fields of the type *Database*.

Database dynamic fields can be created the same way, as the default dynamic fields are created. For this switch to the *Admin* → *Ticket Settings* → *Dynamic Fields* view. In this mask you can select the *Database* field from the ticket drop down box on the left side. Currently it's not possible to use the *Database* dynamic field in the article context.

- Configuration *General* - *Name*:

Dynamic fields of the type *Database* need a unique name just like other dynamic fields, too. This name has to contain only alphanumeric values. This name will be used for internal handling of the field but will not be displayed.

- Configuration *General* - *Label*:

The label can be individually set and can contain white spaces etc. It will be used as the field label in the different views.

- Configuration *General* - *Field order*:

The field order allows a administrator to change the order of created dynamic fields. If this configuration get changed the general field order will be adjusted and other dynamic fields will be moved one position back.

- Configuration *General* - *Valid*:

To use the dynamic field in the configured OTRS views it has to be set as valid. If the field is set to *invalid* it will disappear from all the configured views but no data will be lost.

3.2.2. Configuration of the External Data Source

Before an external database can be searched and the results be saved at the ticket through the dynamic field, the credentials have to be stored in the configuration of the dynamic field.

Figure 3.17. Dynamic Field Database Admin screen

Dynamic Fields - Ticket: Add Database Field

Dynamic Fields Management | Ticket: Add Database Field

Actions: [Go back to overview](#)

General

• Name: Validity:
Must be unique and only accept alphabetic and numeric characters.

• Label: Field type:
This is the name to be shown on the screens where the field is active.

• Field order: Object type:
This is the order in which this field will be shown on the screens where is active.

Database Field Settings

Possible values:

Name	Label	Datatype	Filter	Searchfield	Listfield
• Name: id	• Label: ID	• Datatype: Integer	Filter:	Searchfield: <input checked="" type="checkbox"/>	Listfield: <input type="checkbox"/>
• Name: node	• Label: Node	• Datatype: Text	Filter:	Searchfield: <input checked="" type="checkbox"/>	Listfield: <input checked="" type="checkbox"/>
• Name: customer	• Label: Customer	• Datatype: Text	Filter:	Searchfield: <input checked="" type="checkbox"/>	Listfield: <input checked="" type="checkbox"/>
• Name: type	• Label: Type	• Datatype: Text	Filter:	Searchfield: <input checked="" type="checkbox"/>	Listfield: <input checked="" type="checkbox"/>

Add value:

Show link:

Here you can specify an optional HTTP link for the field value in Overview and Zoom screens.
 Example: <http://some.example.com/handle?query={% Data.Field1 | uri %}>

Link for preview:

If filled in, this URL will be used for a preview which is shown when this link is hovered in ticket zoom. Please note that for this to work, the regular URL link above needs to be filled in, too.

• Type:

• Server:

Port:

• Database:

• Table / View:

• User:

• Password:

• Identifier:

Must be unique column from the table entered in Table/View.

Multiselect:

CacheTTL:

Searchprefix:

Searchsuffix:

Result Limit:

Case Sensitive:

[Save](#) or [Cancel](#)

Type: The type of the desired database can be selected here. The field supports the default OTRS database types: MySQL, Oracle, PostgreSQL or MSSQL.

- **SID:** This option is only available for Oracle connections and will be shown or hidden automatically. Within this option you have to enter the SID of your Oracle connection.
- **Driver:** This option is only available for ODBC connections and will be shown or hidden automatically. Within this option you have to enter in the hostsystem previously configured ODBC driver to connect to the desired MSSQL database.
- **Server:** The database host (hostname or IP-Address).
- **Port:** The port of the database server.
- **Database:** Defines the desired target database of the DBMS. This database will be used for queries.
- **Table / View:** This table or view will be used for the queries.
- **User:** The username for the database connection.
- **Password:** The user password for the database connection.
- **Identifier:** This select box will be automatically filled through *Possible Values (description below)*. This field represents the value which will be stored in the dynamic field.
- **Multiselect:** If this field is selected, it will be possible to store more than one value to the dynamic field. Those values will be stored comma separated.
- **CacheTTL:** This value defines the period of validity of the database cache in seconds. Equal queries to the database will be answered through the cache (local filesystem) within this period instead of asking the database again.
- **Searchprefix:** This value will be put in the front of every search term while using the autocompletion to search the database. Wildcard characters are supported as well. The

searchprefix will be ignored during the detailed search, but it is still possible to use wildcard characters in those masks.

- *Searchsuffix*: This value will be put in the end of every search term while using the autocompletion to search the database. Wildcard characters are supported as well. The searchsuffix will be ignored during the detailed search, but it is still possible to use wildcard characters in those masks.
- *Result Limit*: The entered integer value defines the maximum amount of allowed results during a database search. This includes the autocompletion search as well as the detailed search.
- *Case Sensitive*: If this field is selected, case-sensitivity will take effect on searches.
- *Possible values*: As already explained the possible values will fill up the identifier field automatically, which defines the value that will be stored in the dynamic field. Possible values can be created as much as needed (or at least as many table columns as the database table has). The possible values defines the database columns to search in. It is possible to set the column name, a description (label) the field should have, the needed data type and if the field should be a search- or listfield.
- *Name*: The exact name of the database column which will be requested through the database queries.
- *Description*: The label of the field which will be displayed in the detailed search.
- *Datatype*: The data type which will be stored in the dynamic field. Possible values: TEXT, INTEGER or DATE.
- *Filter*: With the filter field, it is possible to choose a ticket attribute or a dynamic field as a filter for the related column. If the dynamic field is bound to a related ticket, the attributes will be used for the filter mechanism, otherwise the filters will be ignored. If filter will be configured to a table column, only search results matching to the search term and the related ticket attribute on exactly the configured column will be displayed.
- *Searchfield*: Indicates if a field should be included in the search requests.
- *Listfield*: Indicates if a field should be displayed in the results.

3.2.3. Screen Configuration

Dynamic fields of type *Database* have to be activated for the several masks in which they should be displayed like the other types of dynamic fields.

This can be done through *Admin* → *System Administration* → *SysConfig*, in which *Ticket* must be selected on the left hand side.

For every interface area (Frontend), in which the dynamic field of type *Database* should be displayed, the admin has to configure it to fit his needs. Examples:

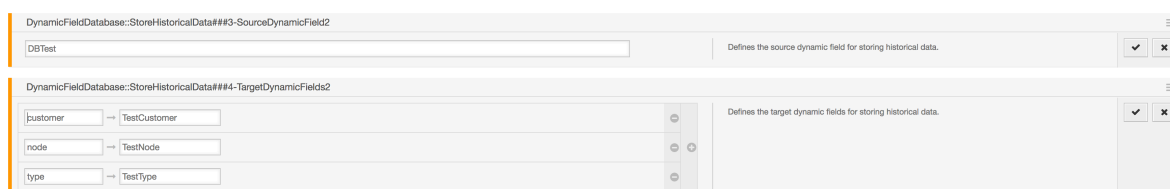
- `Frontend::Agent::Ticket::ViewZoom` for the ticket zoom view.
- `Frontend::Agent::Ticket::ViewPhoneNew` for new phone tickets.
- `Frontend::Agent::Ticket::ViewEmailNew` for new email tickets.
- In each of these view configurations is an entry called `Ticket::Frontend::AgentTicket*###DynamicField`. This configuration defines which dynamic field(s) should get displayed in this view. To add a dynamic field, the internal name of the field has to be filled

in the *Key* field. The field *Value* can take the values 0 (deactivated), 1 (active) and 2 (active and mandatory).

3.2.4. Storage of Historical Data

This feature offers a functionality to store historical data. For this to work it's necessary to activate and set SysConfig options, as visibly in the following screenshot:

Figure 3.18. DynamicField Database - Historical data settings



In the configuration option for the *SourceDynamicField* it's needed to fill in the already created dynamic (Database) field name, which will be used to gather the historical data. In the related option *TargetDynamicField* the field(s) *Key* have to be filled with the table columns of the connected external database, which will be readout. For every column the related target dynamic field has to be configured in the field *content*. The gathered data will be saved in these dynamic fields.

If the configuration is ready and active, the configured fields will be readout from the external database, since the source field gets a new value via the configured masks. The data will be searched by it's stored identifier via an event module and the found values will be stored in the target dynamic fields.

3.2.5. Sysconfig Settings

`AutoComplete::Agent###DynamicFieldDatabaseSearch`

Group: Framework, Subgroup: Frontend::Agent

Defines the config options for the autocompletion feature.

`Ticket::EventModulePost###950-StoreHistoricalData`

Group: Ticket, Subgroup: Core::Ticket

Updates dynamic fields, if configured ones will be updated.

`DynamicFieldDatabase::StoreHistoricalData###1-SourceDynamicField1`

Group: OTRSBusiness, Subgroup: Core

Defines the source dynamic field for storing historical data.

`DynamicFieldDatabase::StoreHistoricalData###2-TargetDynamicFields1`

Group: OTRSBusiness, Subgroup: Core

Defines the target dynamic fields for storing historical data.

`DynamicFieldDatabase::StoreHistoricalData###3-SourceDynamicField2`

Group: OTRSBusiness, Subgroup: Core

Defines the source dynamic field for storing historical data.

DynamicFieldDatabase::StoreHistoricalData###4-TargetDynamicFields2

Group: OTRSBusiness, Subgroup: Core

Defines the target dynamic fields for storing historical data.

3.3. Usage

An exemplary usage of DynamicField Database is as follows:

3.3.1. Searching and Saving Datasets - Autocompletion

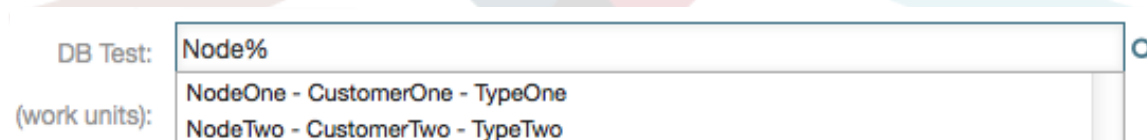
After the created dynamic fields are activated in the well known masks (like ViewPhone-New, ViewEmailNew) a new text field appears with the name, the dynamic field got in the configuration. In this field it is possible to input searchterms and therefore execute a search over all configured database fields. Otherwise do a click on the link *Detailed search* and start a detailed search in which the fields to search in are selected explicitly.

Figure 3.19. DynamicField Database - Test field



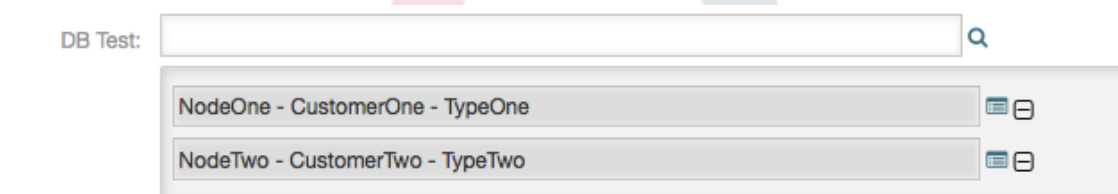
Since search terms are typed in into the text field, a database search will be started over the configured columns and the result will displayed via an autocompletion below the text field. The more exact the search term is, the more exact will be the result (less result entries).

Figure 3.20. DynamicField Database - Autocomplete feature



If the wished value will be displayed in the results, it can be selected via a mouse click or via the keyboard and therefore be added to the dynamic field results.

Figure 3.21. DynamicField Database - Selected items



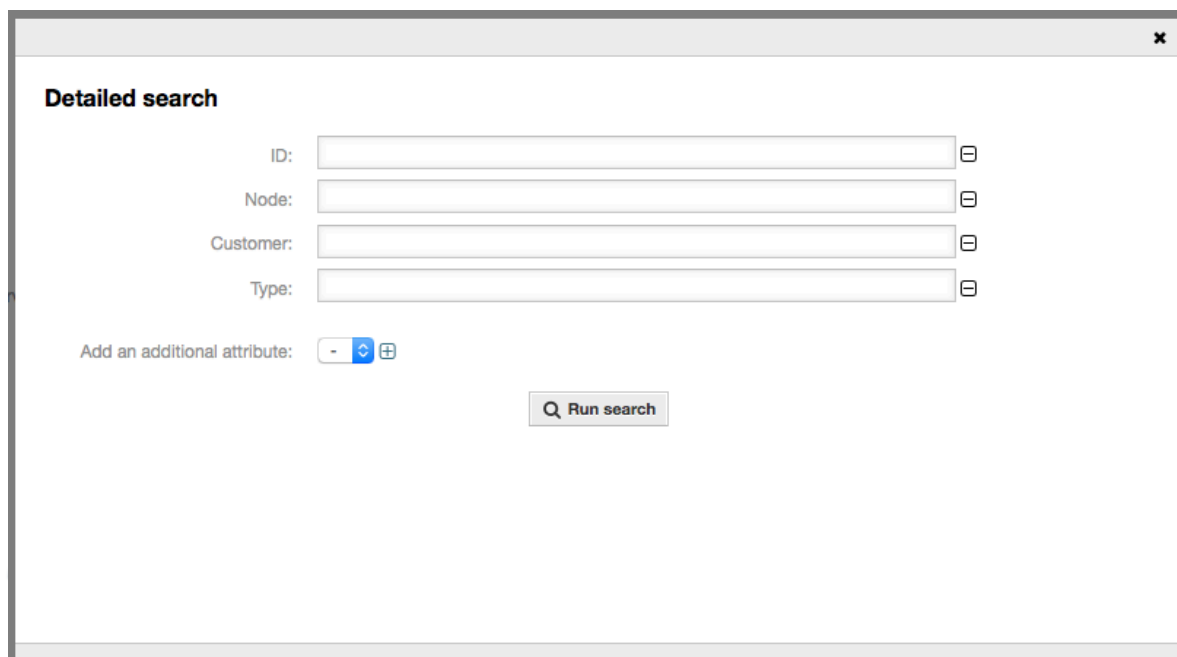
Via the link *Details* a popup screen can be accessed, which offers detailed information about the whole result row. This information includes the line headers and the data. This information can be used to get an overview about the rest (of the not configured) columns or to compare data.

The added result entries can be removed via the minus button.

3.3.2. Searching and Saving Datasets - Detailed Search

The link *Detailed search* opens a new modal dialog to start a new database search. In this mask it is possible to select the fields to search on explicitly.

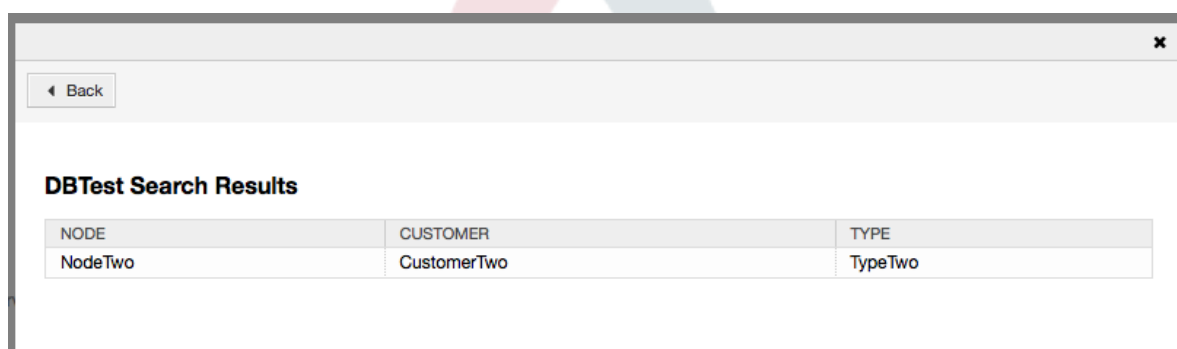
Figure 3.22. DynamicField Database - Detailed search



By default the first available field is activated, but it's also possible to remove available fields or add additional ones. Only activated and filled fields are considered for the search. Wildcard characters '*' are allowed in every single field.

The database search will be executed via the *Start search* button and the results will be tabular displayed. If the search was successful, the results will be listed and one of the entries can be selected via a mouse click. The value will be added to the list of saved values afterwards.

Figure 3.23. DynamicField Database - Detailed search result



NODE	CUSTOMER	TYPE
NodeTwo	CustomerTwo	TypeTwo

Independent of using the autocompletion or the detailed search, every single result can just selected ones. If an agent tries to select a value multiple times, a related warning message is displayed.

4. The Dynamic Field *Web Service*

This feature implements a dynamic field of the type *Web Service*.

4.1. Description

This feature implements a generic dynamic field type that gathers its selectable options from an external system using a web service. The response from the external system defines the possible options to be displayed, and they could vary depending on the data that is sent in the request.

Normally when a field is changed in a screen (e.g. the ticket priority in the *New phone ticket* screen) the values of other fields could be updated. That is the case with this type of dynamic fields, as they could also include all screen field values in the request and the remote server could potentially return completely different values depending on input.

Additionally if the dynamic field source object already exists (e.g. a ticket, and the field is set in the *Free Fields* screen), the details of the already created ticket are also included in the request.

4.2. Configuration

To make this feature work it is necessary to have the following components:

4.2.1. A web service (at least configured as requester)

To create a new *web service dynamic* field it is necessary to have already a working web service, it requires to have at least one invoker of the type `Generic::PassThrough`. This invoker will be called to fetch the data from the remote server.

The original data that it is sent in a request is similar to the following example.

```
{
  DynamicFieldID => 123,
  DynamicFieldLabel => 'NameX',
  DynamicFieldName => 'NameX',
  DynamicFieldValue => 'Value',
  Form => {
    # Form fields
    # ...
  },
  Ticket => {
    # Ticket attributes
    # ...
  },
  DynamicField => {
    NameX => 'Value'
    NameY => [ 'Value' ],
  },
  UserID => 123,
},
```

The `Form` section contains the fields in the current form in the web browser. This information changes as the screen is filled in.

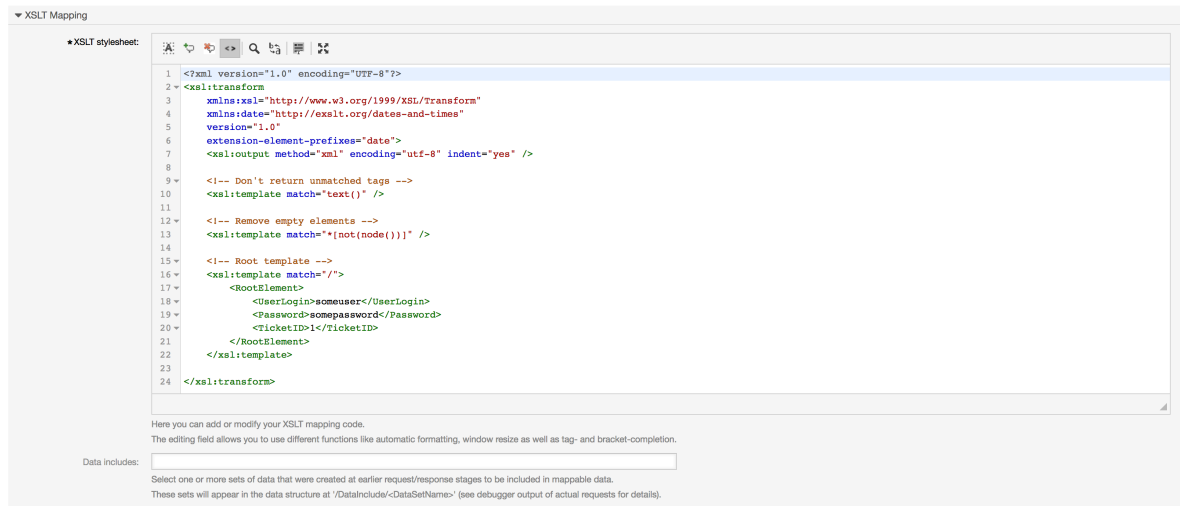
`Ticket` section (or another source object, e.g. `CustomerUser`), contains the attributes of the object where the dynamic field belongs. For example in *New phone ticket* screen the section is empty as the ticket is not created yet, but in *Free Fields* screen it contains the information of the current ticket.

`DynamicField` section contains all non empty values of all configured dynamic fields for the current object.

In most cases the data that the remote server requires will be very different from the data provided, so it is highly recommended to use a mapping module for the outgoing data, to format it specifically for the remote server call.

The following example shows an XSLT mapping that discards any data and sets a fixed UserLogin, Password and TicketID (as needed for a TicketGet operation).

Figure 3.24. Dynamic Field Web Service - Outgoing mapping example



```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <xsl:transform
3   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
4   xmlns:date="http://exslt.org/dates-and-times"
5   version="1.0"
6   extension-element-prefixes="date">
7   <xsl:output method="xml" encoding="utf-8" indent="yes" />
8
9   <!-- Don't return unmatched tags -->
10  <xsl:template match="text()" />
11
12  <!-- Remove empty elements -->
13  <xsl:template match="*[not(node())]" />
14
15  <!-- Root template -->
16  <xsl:template match="/">
17    <RootElement>
18      <UserLogin>someuser</UserLogin>
19      <Password>somepassword</Password>
20      <TicketID>1</TicketID>
21    </RootElement>
22  </xsl:template>
23
24 </xsl:transform>

```

Here you can add or modify your XSLT mapping code.
The editing field allows you to use different functions like automatic formatting, window resize as well as tag- and bracket-completion.

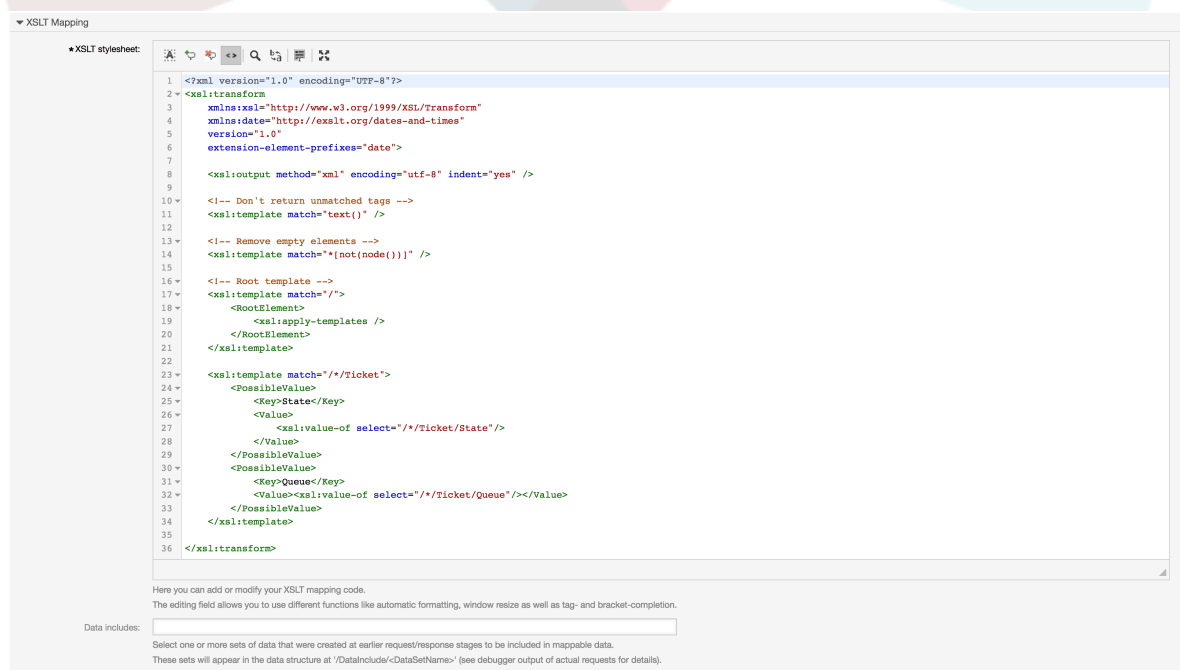
Data includes:

Select one or more sets of data that were created at earlier request/response stages to be included in mappable data.
These sets will appear in the data structure at /Data/include/<DataSetName> (see debugger output of actual requests for details).

The response from the server can also be very different, so in this case is also very recommended to use a mapping module for the incoming data in order to be able to process the information. The response must be a list of Key and Value elements.

The following example shows an XSLT mapping that converts the results from a Ticket-Get operation response form the remote server, extracting and formatting the state and queue as needed to be used as options for the dynamic field.

Figure 3.25. Dynamic Field Web Service - Incoming mapping example



```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <xsl:transform
3   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
4   xmlns:date="http://exslt.org/dates-and-times"
5   version="1.0"
6   extension-element-prefixes="date">
7   <xsl:output method="xml" encoding="utf-8" indent="yes" />
8
9   <!-- Don't return unmatched tags -->
10  <xsl:template match="text()" />
11
12  <!-- Remove empty elements -->
13  <xsl:template match="*[not(node())]" />
14
15  <!-- Root template -->
16  <xsl:template match="/">
17    <RootElement>
18      <xsl:apply-templates />
19    </RootElement>
20  </xsl:template>
21
22  <xsl:template match="*/Ticket">
23    <PossibleValue>
24      <Key>State</Key>
25      <Value>
26        <xsl:value-of select="*/Ticket/State"/>
27      </Value>
28    </PossibleValue>
29    <PossibleValue>
30      <Key>Queue</Key>
31      <Value><xsl:value-of select="*/Ticket/Queue"/></Value>
32    </PossibleValue>
33  </xsl:template>
34
35 </xsl:transform>

```

Here you can add or modify your XSLT mapping code.
The editing field allows you to use different functions like automatic formatting, window resize as well as tag- and bracket-completion.

Data includes:

Select one or more sets of data that were created at earlier request/response stages to be included in mappable data.
These sets will appear in the data structure at /Data/include/<DataSetName> (see debugger output of actual requests for details).

The following web service definition (importable YAML file) can be used for testing the field, but the endpoint must be adapted to match current system. This web service acts

as requester and provider and it always returns the State and Queue from TicketID 1, as possible values to the field.

```

---
Debugger:
  DebugThreshold: debug
  TestMode: '0'
Description: Dynamic Field Web Service Test
FrameworkVersion: 6.0.x git
Provider:
  ErrorHandling: {}
  ErrorHandlingPriority: []
  Operation:
    TicketGet:
      Description: ''
      IncludeTicketData: ''
      MappingInbound: {}
      MappingOutbound: {}
      Type: Ticket::TicketGet
  Transport:
    Config:
      AdditionalHeaders: ~
      MaxLength: '100000000'
      Namespace: http://www.otrs.org/TicketConnector/
      RequestNameFreeText: ''
      RequestNameScheme: Plain
      ResponseNameFreeText: ''
      ResponseNameScheme: Response
    Type: HTTP::SOAP
RemoteSystem: ''
Requester:
  ErrorHandling: {}
  ErrorHandlingPriority: []
  Invoker:
    TicketGet:
      Description: Get possible values from the other side.
      Events: []
      MappingInbound:
        Config:
          Template: "<?xml version='1.0' encoding='UTF-8'>\r\n<!--\r\nCopyright
(C) 2001-2018 OTRS AG, https://otrs.com/\r\nThis software comes with ABSOLUTELY
NO WARRANTY. For details, see\r\nthe enclosed file COPYING for license
information (GPL). If you\r\n did not receive this file, see https://www.gnu.org/
licenses/gpl-3.0.txt.\r\n-->\r\n\r\n<!--
DOCUMENTATION\r\n\r\n* Example XML Input *\r\n<RootElement>\r\n    ...</>\r\n</
RootElement>\r\n\r\n\r\n\r\n*
Example XML Output *\r\n<RootElement>\r\n    <PossibleValues>\r\n
<Key>???

```

```

MappingOutbound:
  Config:
    Template: "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\r\n<!--\r\nCopyright
(C) 2001-2018 OTRS AG, https://otrs.com/\r\n\r\nThis software comes with
ABSOLUTELY NO WARRANTY. For details, see\r\nthe enclosed file COPYING
for license information (GPL). If you\r\ndid not receive this file, see
https://www.gnu.org/licenses/gpl-3.0.txt.\r\n-->\r\n\r\n<!-- DOCUMENTATION\r\n\r
\n*
  Example XML Input *\r\n<RootElement>\r\n    ... \r\n</RootElement>\r\n\r\n\r\n*
  Example XML Output *\r\n<RootElement>\r\n    <PossibleValues>\r\n
<Key>???
```

```

</Key>\r\n
  \    <Value>???
```

```

</Value>\r\n    </PossibleValues>\r\n    <PossibleValues>\r\n
  \    <Key>???
```

```

</Key>\r\n    <Value>???
```

```

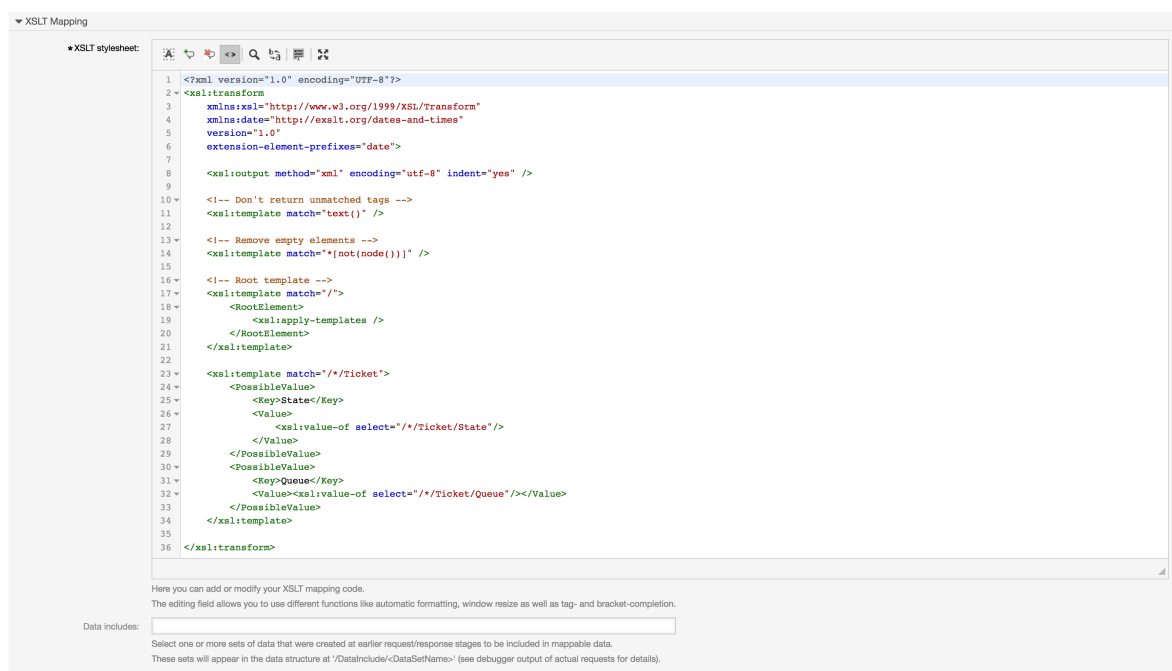
</Value>\r\n    </PossibleValues>\r\n
  \    ... \r\n</RootElement>\r\n\r\n-->\r\n\r\n<xsl:transform\r\n    xmlns:xsl=
"http://www.w3.org/1999/XSL/Transform" \r\n
  \    xmlns:date="http://exslt.org/dates-and-times" \r\n    version="1.0" \r\n
  \    extension-element-prefixes="date">\r\n    <xsl:output method="xml"
encoding="utf-8" indent="yes" />\r\n\r\n    <!-- Don't return unmatched
tags -->\r\n    <xsl:template match="text()" />\r\n\r\n    <!-- Remove
empty elements -->\r\n    <xsl:template match="*[not(node())]" />\r\n\r\n
  \    <!-- Root template -->\r\n    <xsl:template match="/">\r\n
<RootElement>\r\n
  \        <UserLogin>someuser</UserLogin>\r\n
<Password>somepassword</Password>\r\n
  \        <TicketID>1</TicketID>\r\n            </RootElement>\r\n    </
xsl:template>\r\n\r\n</xsl:transform>"
  Type: XSLT
  Type: Generic::PassThrough
  Transport:
  Config:
    Encoding: ''
    Endpoint: http://localhost/otrs/nph-genericinterface.pl/Webservice/
GenericConfigItemConnectorSOAP
  NameSpace: http://www.otrs.org/TicketConnector/
  RequestNameFreeText: ''
  RequestNameScheme: Plain
  ResponseNameFreeText: ''
  ResponseNameScheme: Response
  SOAPAction: Yes
  SOAPActionSeparator: '#'
  SSL:
    SSLProxy: ''
    SSLProxyPassword: ''
    SSLProxyUser: ''
  Type: HTTP::SOAP
  UseMappedData: '1'

```

4.2.2. A web service dynamic field definition

After the proper configuration of web services, it is possible to configure dynamic fields, that use such web services.

Figure 3.26. Dynamic Field Web Service - Definition Example



```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <xsl:transform
3   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
4   xmlns:date="http://exalt.org/dates-and-times"
5   version="1.0"
6   extension-element-prefixes="date">
7
8   <xsl:output method="xml" encoding="utf-8" indent="yes" />
9
10  <!-- Don't return unmatched tags -->
11  <xsl:template match="text()" />
12
13  <!-- Remove empty elements -->
14  <xsl:template match="*[not(node())]" />
15
16  <!-- Root template -->
17  <xsl:template match="/">
18    <RootElement>
19      <xsl:apply-templates />
20    </RootElement>
21  </xsl:template>
22
23  <xsl:template match="*/Ticket">
24    <PossibleValue>
25      <key>State</Key>
26      <Value>
27        <xsl:value-of select="*/Ticket/State"/>
28      </Value>
29    </PossibleValue>
30    <PossibleValue>
31      <key>Queue</Key>
32      <Value><xsl:value-of select="*/Ticket/Queue"/></Value>
33    </PossibleValue>
34  </xsl:template>
35
36 </xsl:transform>

```

Here you can add or modify your XSLT mapping code.
The editing field allows you to use different functions like automatic formatting, window resize as well as tag- and bracket-completion.

Data includes:

Select one or more sets of data that were created at earlier request/response stages to be included in mappable data.
These sets will appear in the data structure at /Data/include/<DataSetName> (see debugger output of actual requests for details).

Below is a description of the dynamic field options.

Web service

The configured web service whose invocers will be triggered when a dynamic field is displayed.

Invoker

The invoker that is used to send requests to external systems. Within this field, just invocers of type `Generic::PassThrough` will be displayed.

Multiselect

A drop-down menu to determine if the displayed dynamic field should act as a multi-select field, instead of a drop-down field.

Cache TTL

A cache time to live value, that contains a value (in minutes). If the value is 0 or empty, no caching will be active.

This cache is to prevent unnecessary requests to the remote server using the same values.

Add empty value

Defines if it is possible to save an empty value in the field.

Tree view

This option activates the tree view of possible values, if they are supplied in the correct format.

Translatable values

Defines if the displayed possible values will be translated to the current user language.

4.2.3. Screen Configuration

Dynamic fields of type *Web Service* have to be activated for the several masks in which they should be displayed like the other types of dynamic fields. The following covers the cases for ticket based dynamic fields.

This can be done through *Admin* → *Administration Overview* → *System Configuration*, then navigate to *Frontend* → *Agent* → *View*.

For every *Ticket** option, in which the dynamic field of type *Web Service* should be displayed, the admin has to configure it to fit his needs. Examples:

- TicketZoom for the ticket zoom view.
- TicketPhoneNew for new phone tickets.
- TicketEmailNew for new email tickets.
- The dynamic field can be added if there exists an entry called `Ticket::Frontend::AgentTicket*###DynamicField`. This configuration defines which dynamic fields should be displayed in this view. To add a dynamic field, the internal name of the field has to be filled in the first field. After confirmation a second field is filled with the display option. Choose *1 - Enabled* to just show the field or *2 - Enabled and required* to make it mandatory.

4.3. Usage

The use of the *Web Service* dynamic field is similar to other dynamic fields like *Dropdown* or *Multiselect*, but users could expect to have different options to choose even without active ACLs due to the change of parameters in the web request as other fields are changed.

5. Import Ready2Adopt Process

5.1. Import

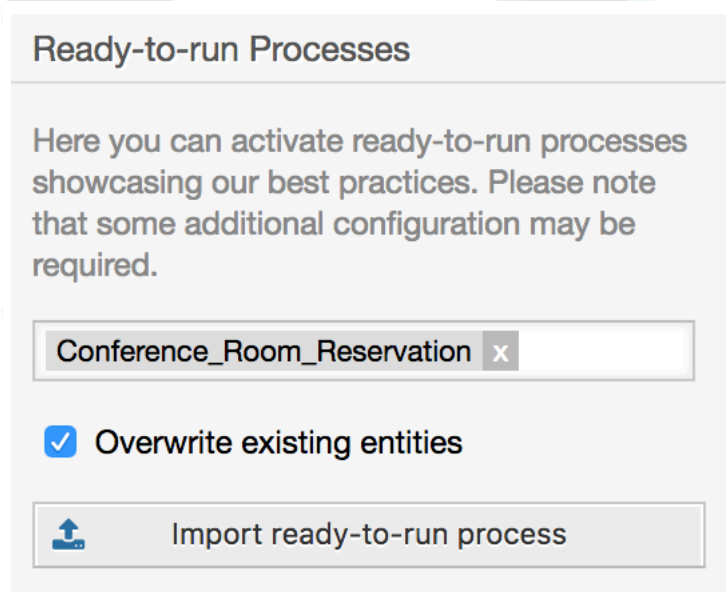
On the *Process Management* screen in the Admin interface you can find an *Ready2Adopt Processes* widget, where you can find some best practice Ready2Adopt processes.

The following Ready2Adopt processes are available in the **OTRS Business Solution™**:

- Conference Room Reservation
- Office Materials Procurement
- Order Request Management
- Request For Leave Management
- Service Design::Availability Management (proactive)
- Service Design::Availability Management (reactive)
- Service Design::Catalogue Management
- Service Design::Continuity Management
- Service Design::Service Design Coordination (individual design)
- Service Design::Service Level Management
- Service Design::Supplier Management

- Service Operation::Access Management
- Service Operation::Event Management
- Service Operation::Incident Management
- Service Operation::Problem Management
- Service Operation::Request Fulfillment Management
- Service Operation::Service Validation and Testing
- Service Transition::Change Evaluation
- Service Transition::Knowledge Management
- Service Transition::Release And Deployment Management
- Service Transition::Service Asset and Configuration Management
- Service Transition::Service Transition Planning and Support
- Start RMA
- Travel Expense

Figure 3.27. Import Ready2Adopt Process Widget



To install a Ready2Adopt process, just select the desired process from the drop-down menu in the *Ready2Adopt Processes* widget and click the *Import Ready2Adopt process* button.

During the import process, OTRS takes care of creating the needed dynamic fields and/or any needed updates to the system configuration.

6. Import Ready2Adopt Web Services

6.1. Import

There are three Ready2Adopt web services available in the **OTRS Business Solution™**:

- BugzillaConnector

- JIRAConector
- OTRSConector

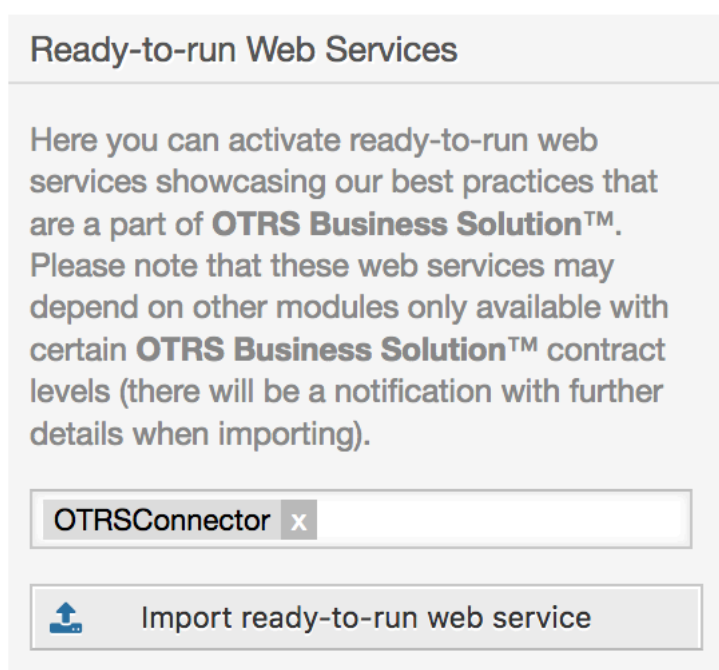
To install one of these web services, go to the *GenericInterface Web Service Management* page and click on the *Add web service* button.

Figure 3.28. Add web service



Locate the *Ready2Adopt Web Services* widget, select one of the provided web services from the drop-down menu, and click the *Import Ready2Adopt web service* button.

Figure 3.29. Import Ready2Adopt Web Service



During the import process, OTRS takes care of creating the needed dynamic fields and/or any needed updates to the system configuration.

Note

The Ready2Adopt web services may require additional feature add-ons to be installed (**OTRSGenericInterfaceInvokerTicket** and **OTRSGenericInterfaceInvokerEventFilter**).

7. SMS Notifications and Notification Web View

7.1. Description

The **OTRS Business Solution™** comes with two new notification features: SMS notifications and the notification web view. Both of them are technically notification methods

besides the existing email method and can be activated for each notification (e.g. *Ticket new note notification* or *Ticket create notification*) on the *Notification Event* screen in the Admin interface.

7.2. Setting up SMS Notifications

Note

Before taking advantage of SMS notification transport method, you must first activate the SMS cloud service.

In order to use SMS as a way of notification for admins and customers, you have to buy an SMS unit package. Please get in touch with sales@otrs.com on this matter. Once you've bought a package, you will be able to look up your current contingent of remaining SMS units on the admin screen for the SMS cloud service (AdminCloudServiceSMS).

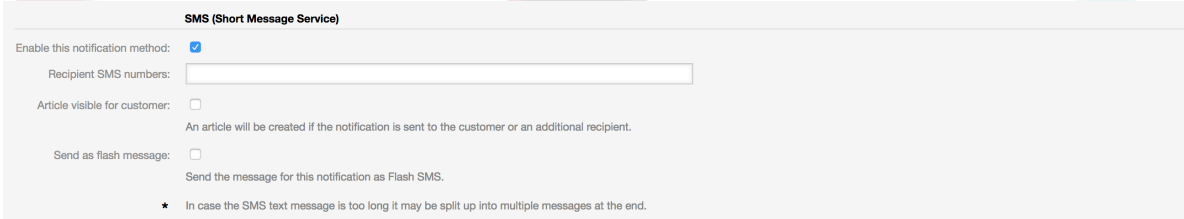
To make SMS available for notifications, you have to enable it for the notifications of your choice in *Notification Event* screen in the Admin interface. Besides the existing *Email* notification method, you can enable SMS notifications by enabling the checkbox as to be seen in the screenshot below.

Note

Please be careful using the *Send by default* checkbox on SMS notifications. Enabling this checkbox will cause this notification to be sent by SMS to all agents with a phone number stored in the configured field. This could cause a high usage of your available SMS units!

On this screen, you can also add more generic phone numbers (e.g. for archiving purposes) which will receive all of the relevant notifications (besides the addressed agents). Please note that all of these recipients will be treated as customers (regarding article creation).

Figure 3.30. Enabling SMS Notifications



SMS (Short Message Service)

Enable this notification method:

Recipient SMS numbers:


Article visible for customer:
An article will be created if the notification is sent to the customer or an additional recipient.

Send as flash message:
Send the message for this notification as Flash SMS.

* In case the SMS text message is too long it may be split up into multiple messages at the end.

If you configured a notification to be visible in the agent preferences, agents will now be able to enable/disable SMS notifications manually on their own for this notification.

Figure 3.31. Agent Preferences SMS Notification Settings

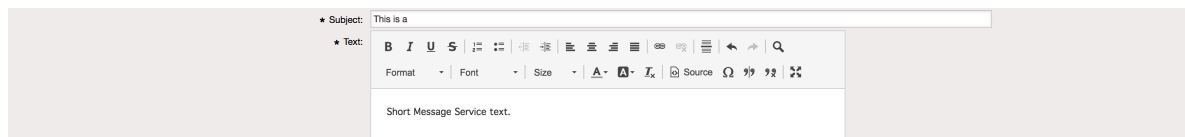


NOTIFICATION	✉	☰	☏
Ticket create notification	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ticket follow-up notification (locked)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ticket follow-up notification (unlocked)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ticket lock timeout notification	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ticket new note notification	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ticket queue update notification	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ticket service update notification	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Choose for which kind of ticket changes you want to receive notifications. Please note that you can't completely disable notifications marked as mandatory.

SMS notification text will be constructed on fly by combining both *Subject* and *Text* fields of the corresponding language. You can of course use supported OTRS tags in these fields, and additionally URLs can be shortened by activating URL shortening option in cloud service configuration.

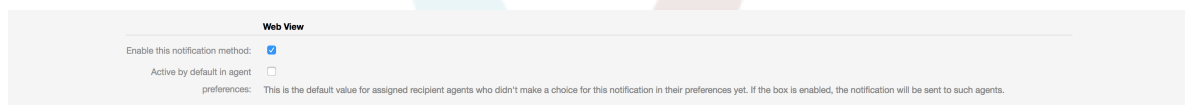
Figure 3.32. SMS Notification Text Fields



7.3. Setting up and Using the Notification Web View

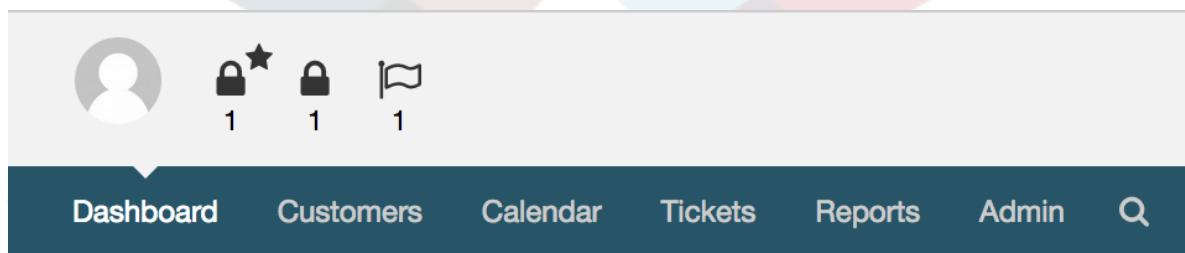
Setting up the notification web view for agents is similar to the setup for SMS notifications as described above.

Figure 3.33. Enabling the Notification Web View



Once you've enabled the notification web view for a certain notification, agents will see a new icon on top of their screen in OTRS (within the toolbar) each time a new notification arrives.

Figure 3.34. Notification Web View Icon on the Toolbar



Clicking this icon, agents will get an overview of all of their notifications and can read and dismiss them from here.

Figure 3.35. Notification Web View Overview

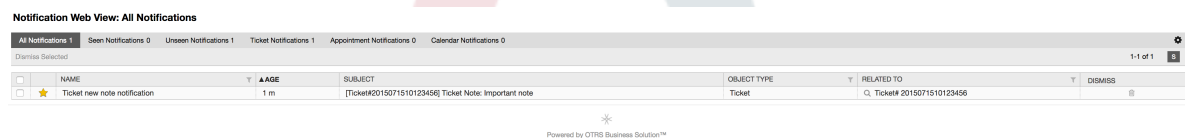
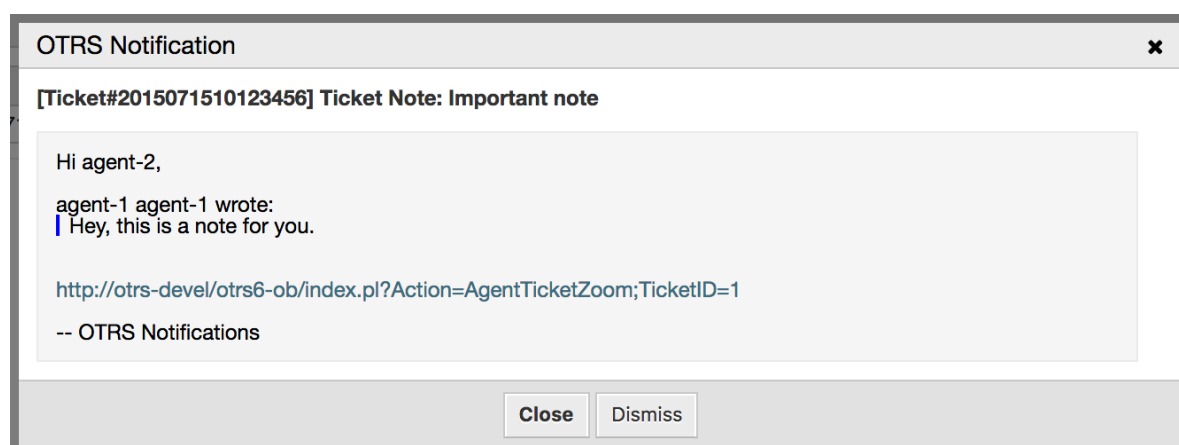


Figure 3.36. Notification Web View Detail



8. The Report Generator

8.1. Description

The report generator allows to combine multiple statistics for sending/downloading. Reports can be configured to be sent automatically on specified times or run manually.

8.2. Configuration

No further configuration needed.

8.3. Usage

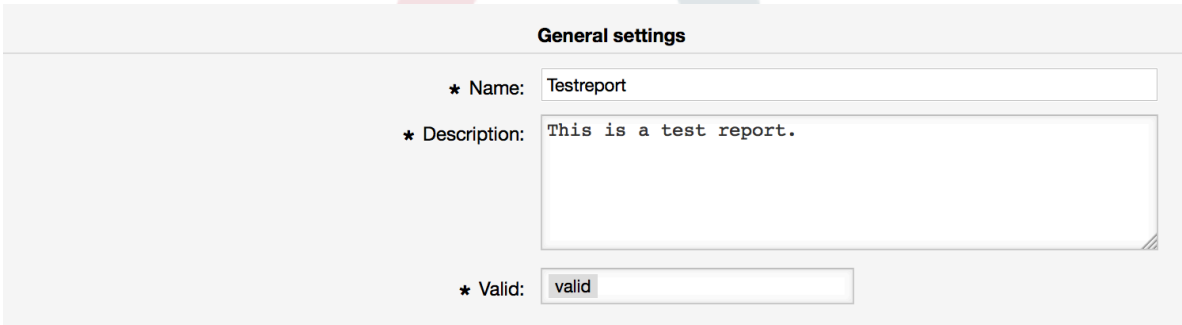
8.3.1. Creating Reports

After you've entered title and description for your report in the first step, you'll be able to set up your report.

8.3.1.1. General Settings

Set up title, description and validity of the report. This information will not be visible in the report.

Figure 3.37. Automatic generation settings



General settings

* Name:

* Description:

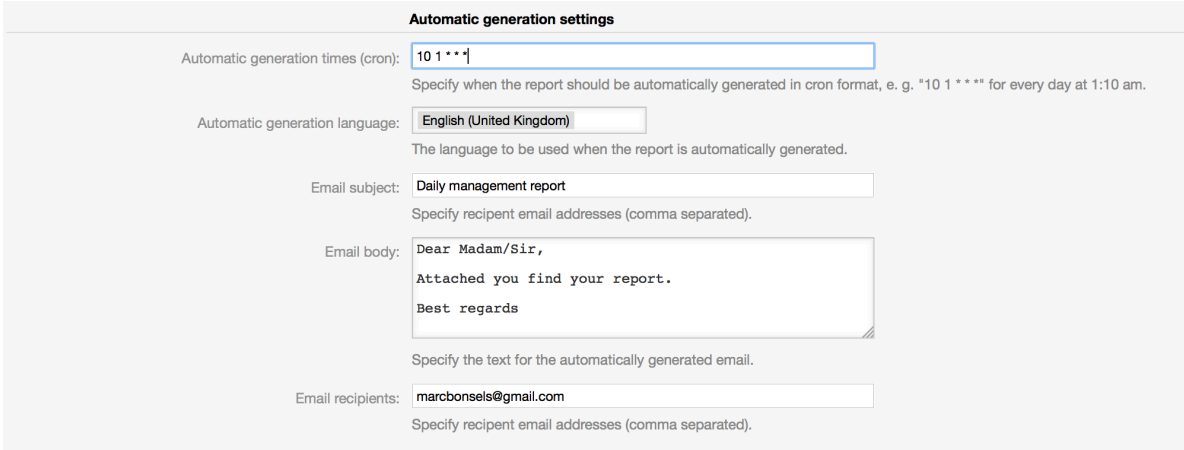
* Valid:

8.3.1.2. Automatic Generation Settings

If the report should be generated automatically (e.g. once per day), you can set up subject, text and recipients for the email which is going to be sent and which will contain the

report as an attachment. You'll also need to add information on how often the report should be sent. The data format is taken from the cron mechanism of unix-based operating systems. For more information on which settings are possible, please refer to official cron documentation sources or [wikipedia](https://en.wikipedia.org/wiki/Cron).

Figure 3.38. Automatic generation settings



Automatic generation settings

Automatic generation times (cron): Specify when the report should be automatically generated in cron format, e. g. "10 1 * * *" for every day at 1:10 am.

Automatic generation language: The language to be used when the report is automatically generated.

Email subject: Specify recipient email addresses (comma separated).

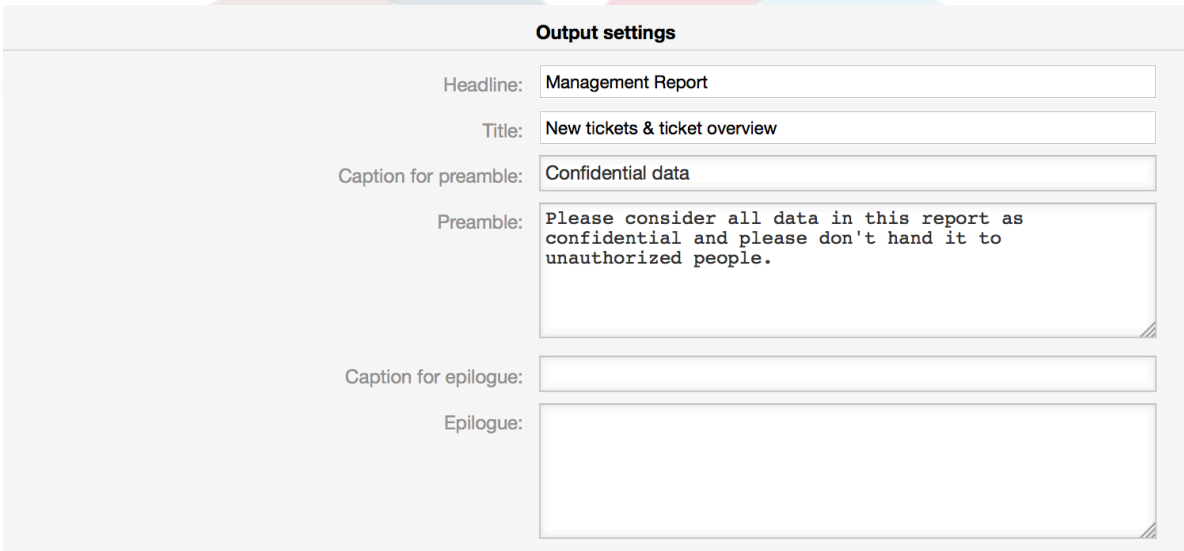
Email body: Specify the text for the automatically generated email.

Email recipients: Specify recipient email addresses (comma separated).

8.3.1.3. Output Settings

In this section, you can set up additional information which should be contained in the report (e.g. a preamble or epilogue). This could e.g. be used for adding copyright or data protection information.

Figure 3.39. Output settings



Output settings

Headline:

Title:

Caption for preamble:

Preamble:

Caption for epilogue:

Epilogue:

8.3.2. Manually Generating Reports

To generate a report, you can use the *Run now* button from the reports overview. You'll be asked for confirmation and afterwards the report will be created and sent immediately. Please find screenshots of a generated report below.

Figure 3.40. Report: Front page

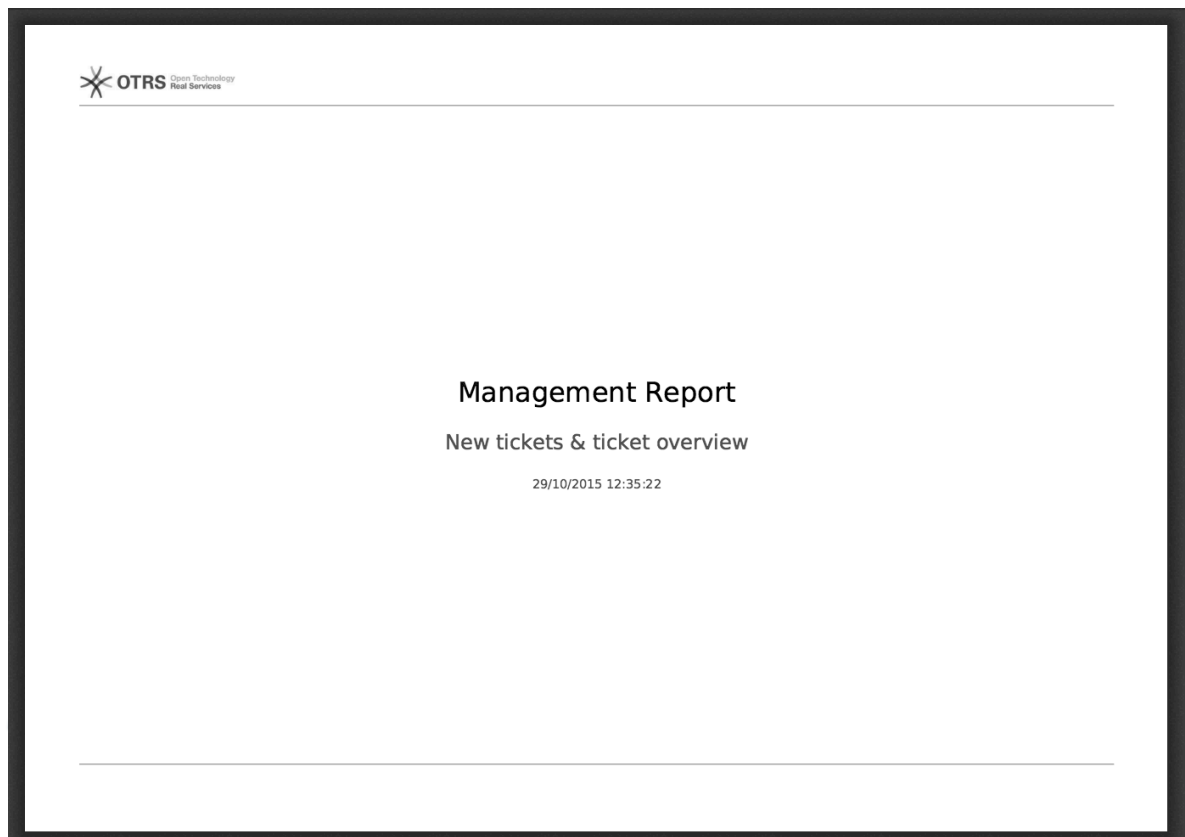


Figure 3.41. Report: Table of contents

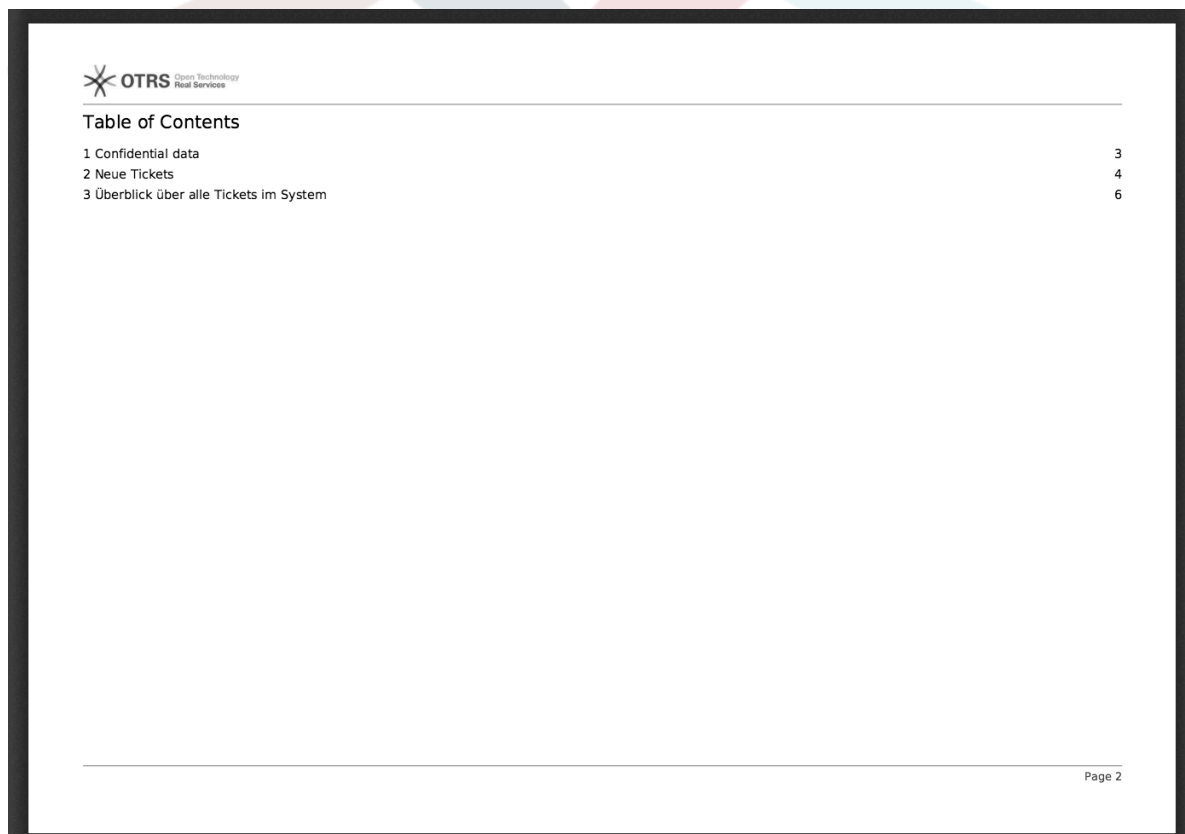


Figure 3.42. Report: Preamble

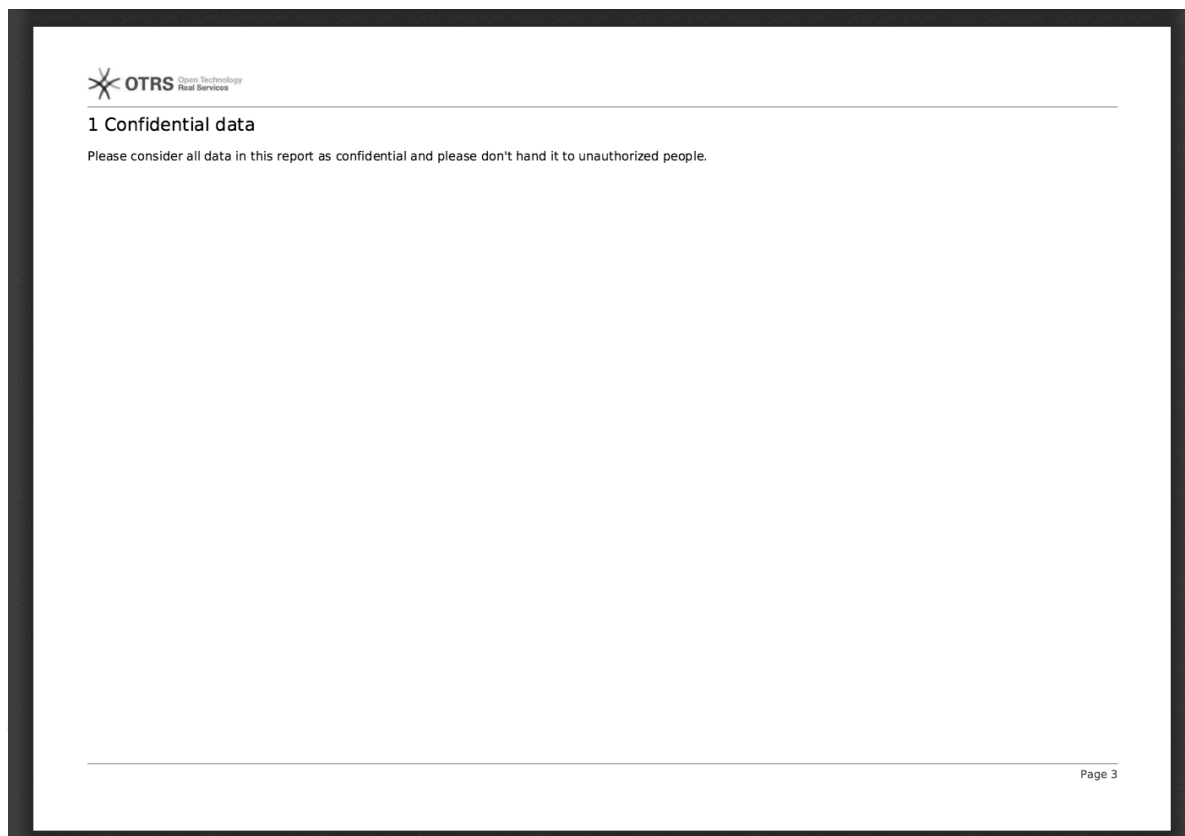
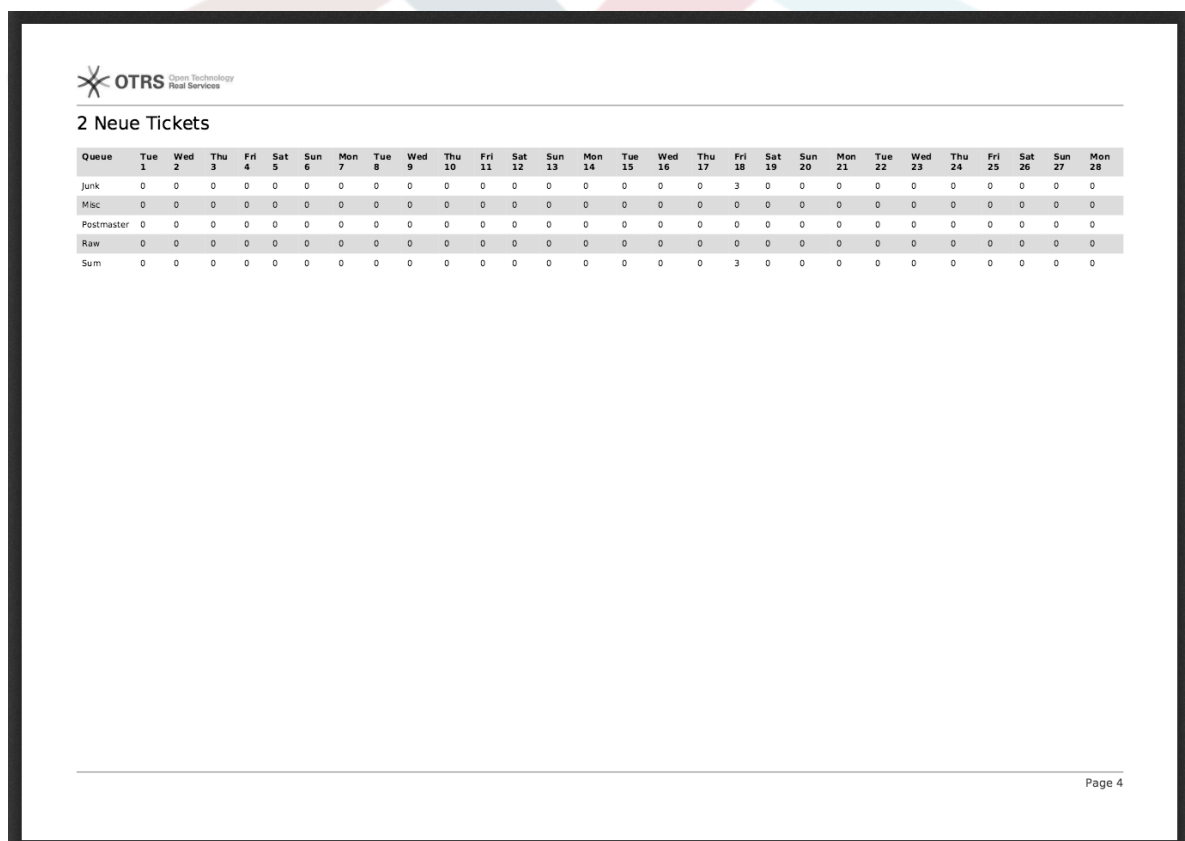


Figure 3.43. Report: Statistics



9. Resource Planning

9.1. Description

Resource planning feature in **OTRS Business Solution™** complements built-in calendar feature of OTRS. It enhances original calendar implementation by introducing resource allocation for appointments. Furthermore, resources can be assigned to separate teams, which in turn can be used to filter resource overview and availability.

9.2. Configuration

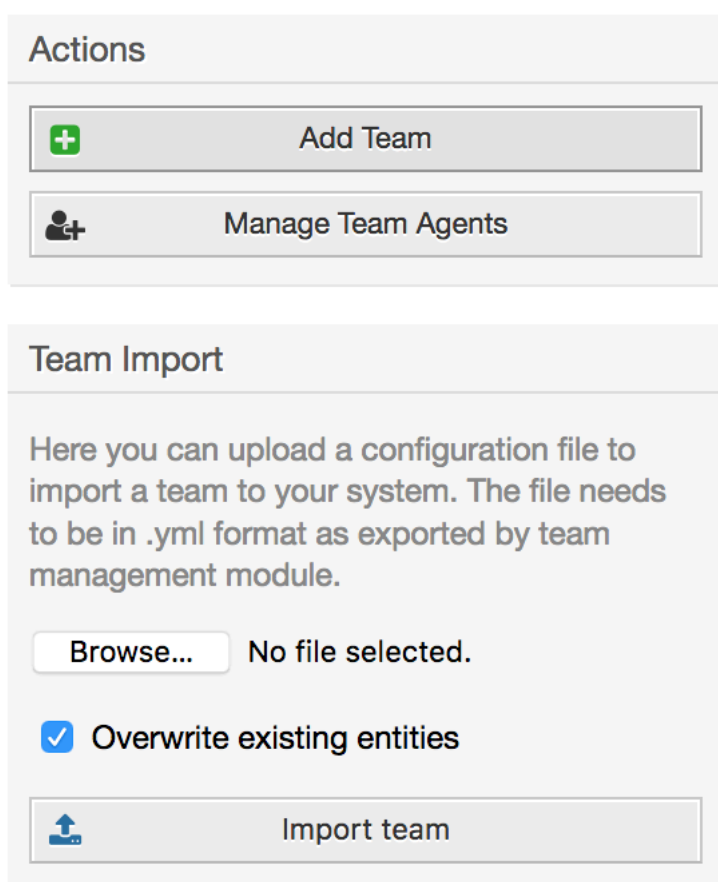
No further configuration needed.

9.3. Usage

9.3.1. Team Management

In order to be able to assign appointments to resources, you must first define teams for those resources.

Figure 3.44. Team Management Add and Import



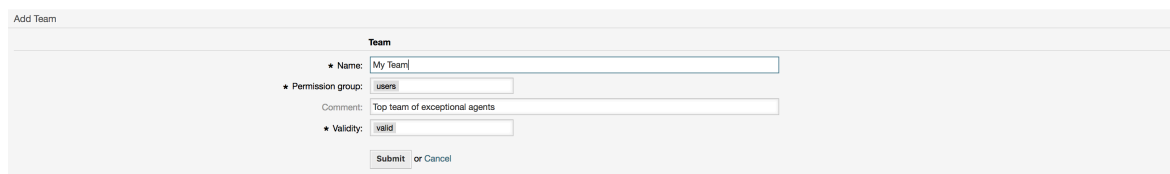
The screenshot displays two main sections of the Team Management interface:

- Actions:** A panel containing two buttons: "Add Team" (with a green plus icon) and "Manage Team Agents" (with a person icon).
- Team Import:** A panel with the following elements:
 - Text: "Here you can upload a configuration file to import a team to your system. The file needs to be in .yml format as exported by team management module."
 - File selection: A "Browse..." button followed by the text "No file selected."
 - Checkbox: A checked checkbox labeled "Overwrite existing entities".
 - Import button: A button with an upload icon and the text "Import team".

Teams can be added or imported from *Manage Teams* screen accessible via top navigation menu. In order to import a team, you must have an exported YML configuration file of the team from an existing OTRS installation.

Adding new team is easy: just define a team name and permission group. Full access to this group is required for anyone to be able to manage the team, its agents and assign them as resources to appointments.

Figure 3.45. Team Management Add and Import



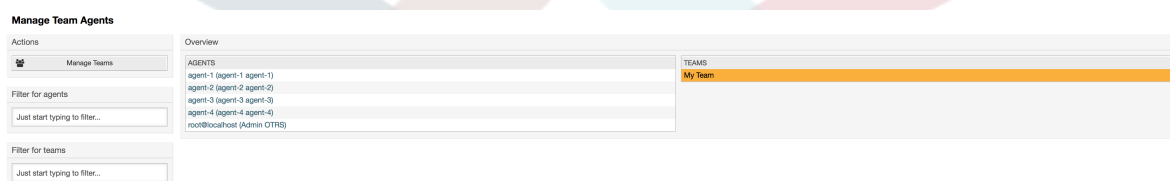
9.3.2. Team Agent Management

Once you have teams set up, you must define which agents belong to which teams. This is done via dedicated screen where you can adjust relations of agents and teams very much like you can do it for agents and groups.

Note

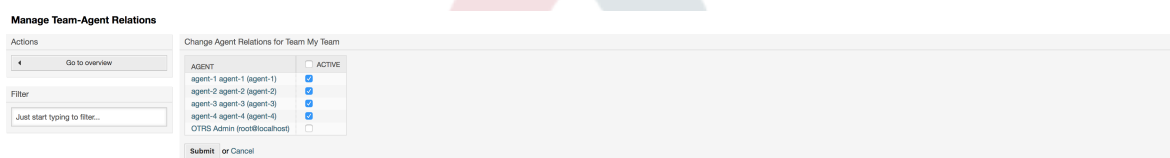
Please note that in order to see agents in this screen, you must define them first via *Admin* → *Agents* screen.

Figure 3.46. Team Agent Management



By choosing a team from this screen, you can assign agents to it, and vice-versa.

Figure 3.47. Team Agent Relations



9.3.3. Assigning Appointments to Resources

Finally, once you have both teams and their agents set up, you can proceed to a calendar overview and start assigning appointments to teams and agents.

Simply add or edit an appointment, and choose both team and agent from the lists under resource. You can do this via any calendar overview, as appointment edit dialog is the same for all of them.

Figure 3.48. Assigning team and agent to an appointment

Appointment
✕

Basic information

* Title:

Description:

Location:

* Calendar:

Resource

Team:

Agent:

Date/Time

Start date: / / :

End date: / / :

All-day:

Repeat:

Notification

Notification:

Link


Ticket:

You can assign multiple teams and agents to a single appointment, but you must always choose a team first. Resulting list of agents is an aggregate of all selected teams, without any duplicates. Choosing an agent from the list is also optional, so you can leave an appointment assigned only to a team.

Figure 3.49. Assigning multiple teams and agents to an appointment

Resource

Team: My Backup Team x My Team x

Agent: Admin OTRS x agent-1 agent-1 x and 3 more... 

Note

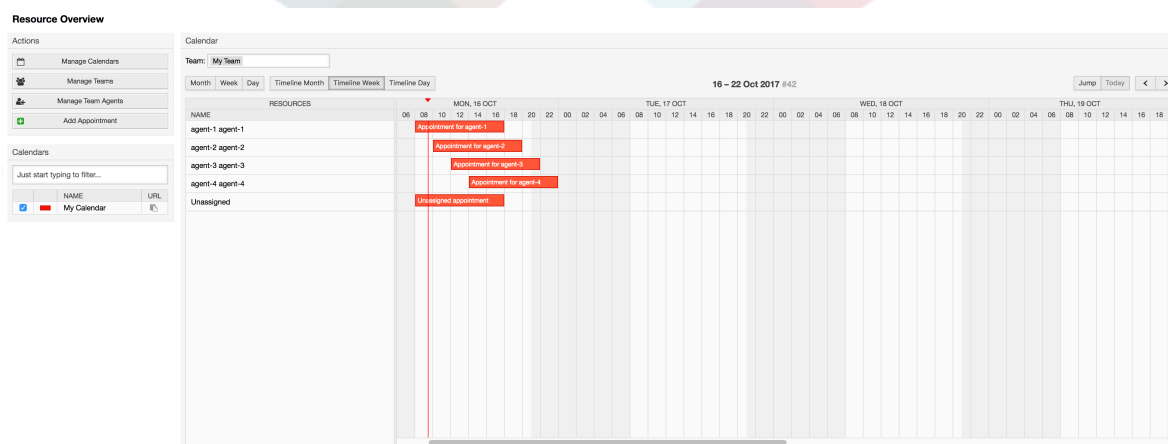
Once you assign an appointment to a team and/or agent, make sure to save it so your changes are applied.

9.3.4. Resource Overview

While it's possible to assign resources to an appointment in any calendar overview, one screen is designed exclusively for allocating resources to appointments: Resource Overview. This screen looks very much like a regular calendar overview, except it has a filter for teams on top and a list of agents for that team (available only in timeline views).

By choosing a team from the top filter, you will be presented with all appointments assigned to it.

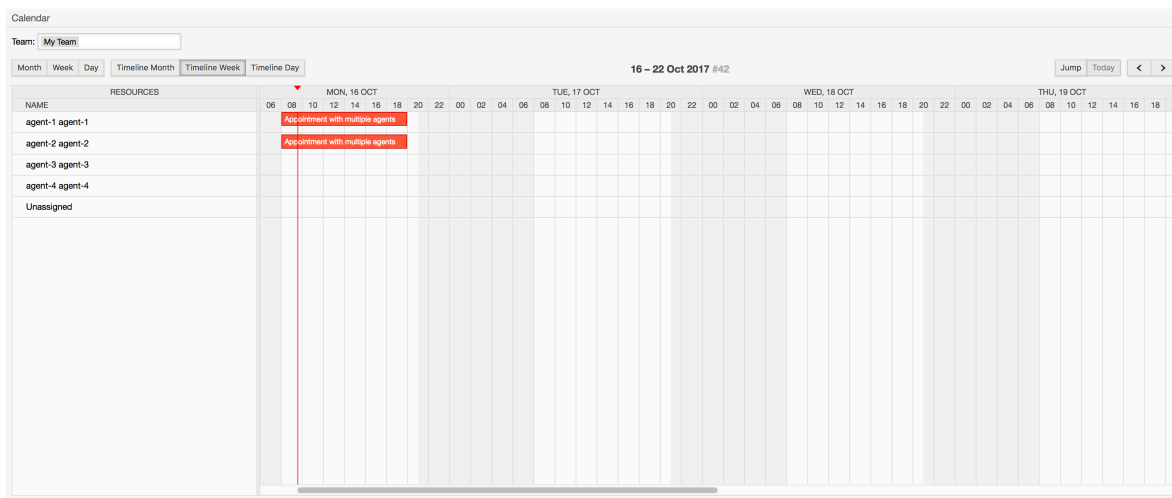
Figure 3.50. Resource Overview



By simply dragging an appointment to a row line with an agent's name in timeline view, it's possible to assign an appointment to this resource. If an appointment does not have an agent assigned to it, it will appear in the unassigned area of the screen.

Appointment with multiple agents will be displayed as duplicated, but it most certainly is not. Dragging one instance of this appointment will move all them in the same fashion.

Figure 3.51. Resource Overview

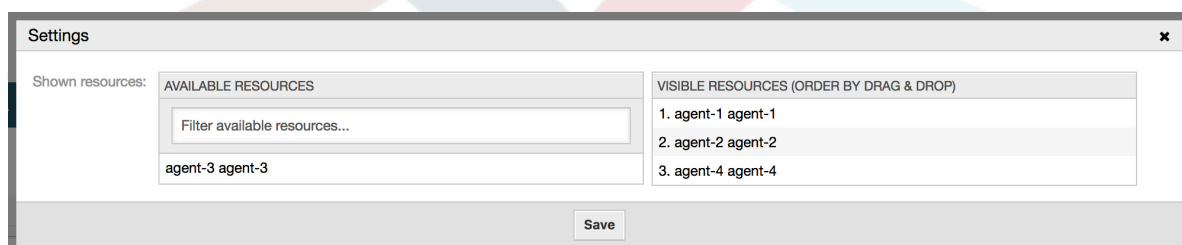


Note

Please note that appointments with unassigned teams or assigned different team than currently selected, will not show up in the resource overview. You will have to switch to a different overview to see these appointments, so make sure to assign them to a correct team first.

By editing settings of the resource overview (via gear icon in top right corner), you can choose which team agents are displayed within the overview.

Figure 3.52. Resource Overview Settings



This setting is user and team specific, and if you have defined an agent filter in this way, a trash icon will be displayed on top of the list of agents so you can remove it easily.

Figure 3.53. Resource Overview Filter Delete

Calendar

Team:

RESOURCES		MON, 10				
NAME		06	08	10	12	1
agent-1 agent-1						
agent-2 agent-2						
agent-4 agent-4						
Unassigned						

Restore default settings

In all other aspects, resource overview screen behaves in the same way as a regular calendar overview.

10. The SLA Field Selection Dialog

10.1. Description

The field selection dialog enables you to configure popup messages customers should see when they select specific SLAs on ticket creation on the customer frontend. These popup messages could contain information about the SLA the customer is about to select and to give them the possibility to re-think their decision.

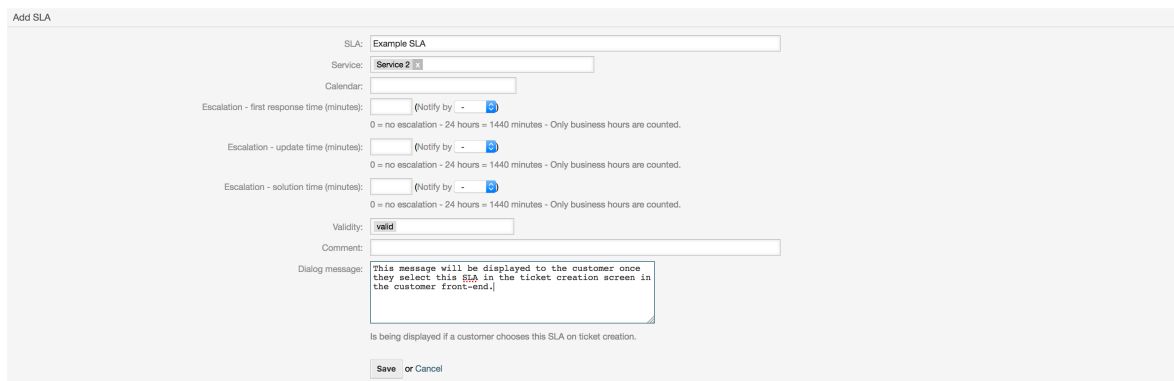
10.2. Configuration

No further configuration needed.

10.3. Usage

10.3.1. Configuring the SLA message

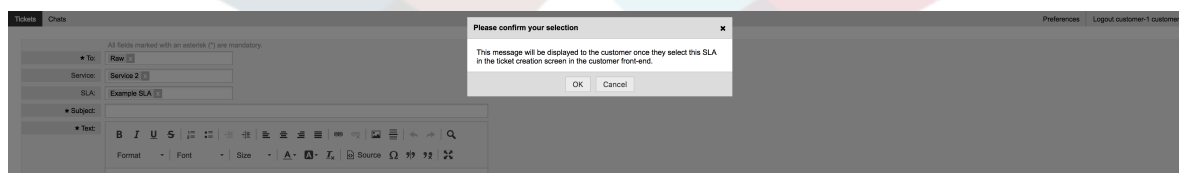
Figure 3.54. SLA configuration



You can configure the message on each SLA by using the SLA administration screen.

10.3.2. How the message looks in the customer front-end

Figure 3.55. SLA message



This is what the customer will see once he selects the SLA this message is configured for. He can either confirm or deny his selection, the latter one will reset the SLA selection afterwards.

11. SMS Communication Channel

11.1. Description

The **OTRS Business Solution™** brings another communication channel to your OTRS instance: SMS (Short Message Service). It allows you to leverage SMS cloud service to send out messages and store them as article in tickets.

11.2. Activating SMS Cloud Service

Before taking advantage of SMS cloud service, you must first activate it. To do so, please visit the *Cloud Services* screen in Admin interface, and click on SMS entry in table.

Figure 3.56. Cloud Services Screen in Admin Interface



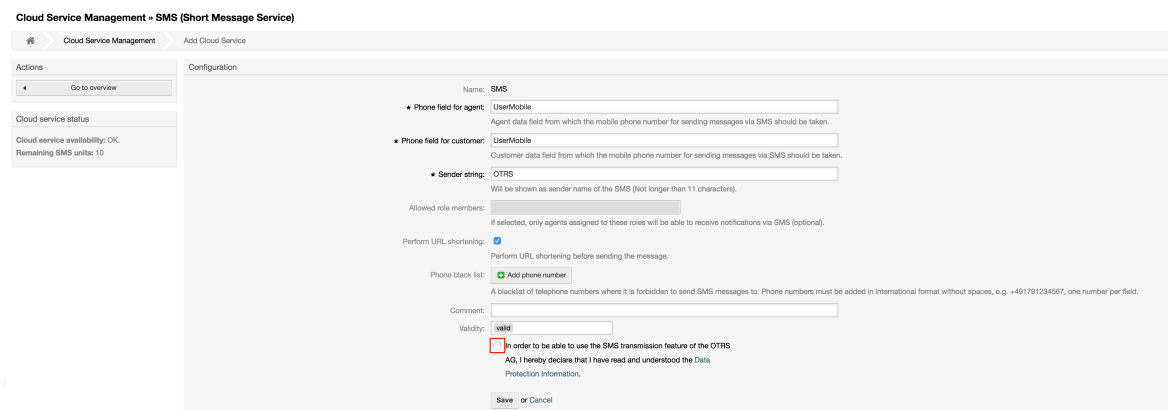
NAME	DESCRIPTION
Support data collector	Configure sending of support data to OTRS Group for improved support.
SMS	This will allow the system to send text messages via SMS.

Note

In order to use SMS cloud service, you have to buy an SMS unit package. Please get in touch with sales@otrs.com on this matter. Once you've bought a package, you will be able to look up your current contingent of remaining SMS units in the sidebar of the cloud service configuration screen.

To activate SMS cloud service, make sure to accept the Data Protection agreement by selecting the checkbox.

Figure 3.57. Activating SMS Cloud Service



On this screen, you can also set up some additional parameters of the SMS cloud service, e.g. the sender string which should be used for the SMS text messages or the agent/customer preference field name which should be used to take the mobile phone number from.

Note

If you change value for the customer preference field, make sure to update the configuration options of the address book in the system configuration to reflect this change:

- CustomerUser::Frontend::AgentCustomerUserAddressBook###SearchParameters###SMS
- CustomerUser::Frontend::AgentCustomerUserAddressBook###DefaultFields###SMS
- CustomerUser::Frontend::AgentCustomerUserAddressBook###ShowColumns###SMS

This will make sure your configuration stays in sync and phone numbers can be found/copied from correct fields.

Allowed role members is an optional white list of agent roles whose members will be able to receive SMS notifications. If undefined, all agents will have a possibility of receiving SMS notifications.

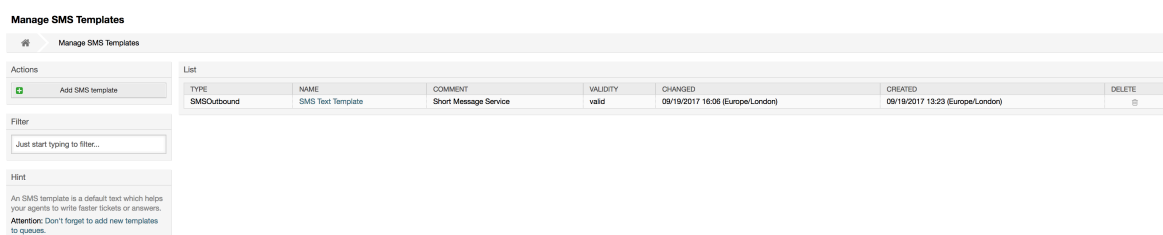
Perform URL shortening checkbox controls the feature of shortening links in SMS texts. If enabled, any link submitted as part of SMS text will be shortened by using OTRS cloud URL shortening service. Links will be shortened only if their length is larger than resulting shortened URL, in order to keep SMS costs down.

By adding specific phone numbers to the *Phone black list*, you can make sure these numbers will never receive any SMS text via cloud service (whether it's a notification or an article). Make sure to input phone numbers in international format, complete with plus sign (+) and country code, without any spaces.

11.3. Configuring SMS Templates

In order to benefit from SMS templates, you must first configure them for specific screens and assign them to correct queues. You can find *SMS template management* screen in the Admin interface.

Figure 3.58. Manage SMS Templates



SMS templates behave similar to regular email templates. You define them first, and then assign to queues with which you would like to use them.

Template Type field determines for which screen the template will be visible:

Answer

Template will be visible for *Reply via SMS article* action.

Create

Template will be visible in *New SMS ticket* screen.

SMSOutbound

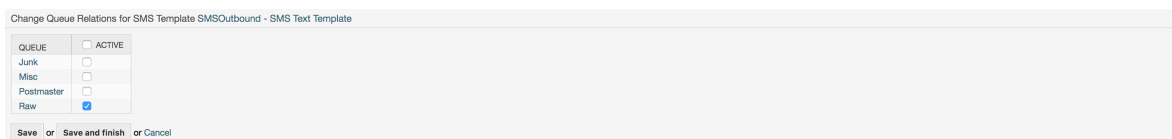
Template will be visible in *SMS Outbound* screen.

Note

Create type templates only supports these smart tags: `<OTRS_CURRENT_*>` and `<OTRS_CONFIG_*>`.

When you define an SMS template, don't forget to add them to queues. You can do this by following link in the sidebar or via *SMS Templates* ↔ *Queues* link in the admin interface.

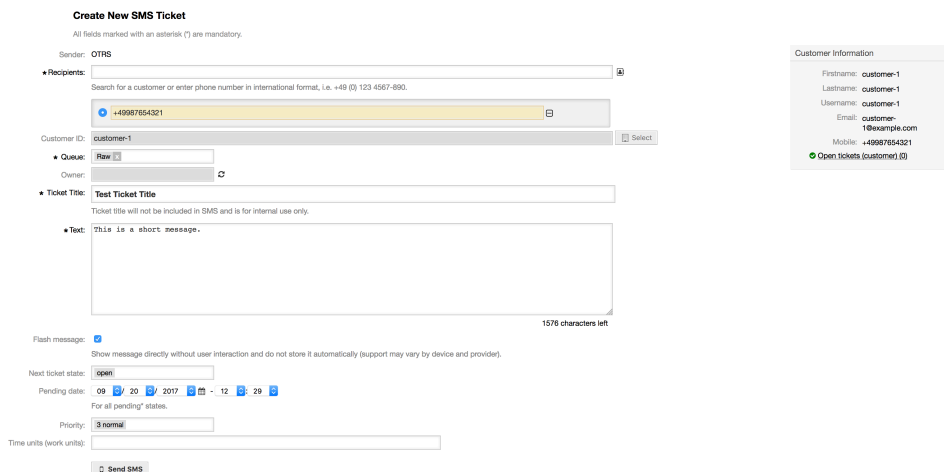
Figure 3.59. Manage SMS Templates



11.4. Creating SMS Tickets

To create a ticket with SMS article as a first one, please go to the *New SMS ticket* screen accessible via *Tickets* menu on the main toolbar.

Figure 3.60. New SMS Ticket Screen



Create New SMS Ticket
 All fields marked with an asterisk (*) are mandatory.

Sender: OTRS

* Recipients:
 Search for a customer or enter phone number in international format, i.e. +49 (0) 123 4567-890.

Customer ID: customer-1

* Queue: Raw
 Owner:

* Ticket Title:
 Ticket title will not be included in SMS and is for internal use only.

* Text:
 1578 characters left

Flash message: Show message directly without user interaction and do not store it automatically (support may vary by device and provider).

Next ticket state:

Pending date: 09/20/2017 - 12:29
 For all pending states.

Priority:

Time units (work units):

Customer Information
 Firstname: customer-1
 Lastname: customer-1
 Username: customer-1
 Email: customer-1@sample.com
 Mobile: +49987654321

This screen behaves similar to new email and phone screens, with some notable exceptions. Instead of email addresses, *Recipients* field accepts only phone numbers in international format, complete with plus sign (+) and country code. You can also search for customer users via this field, provided they have a valid phone number stored in preference field which is configured in cloud service configuration screen.

You may also utilize address book by clicking on small icon next to the recipients field. Search form will allow you to find customer user you are looking for, and add their phone number to the list of recipients.

Note

Ticket Title field will not be included in SMS and is for internal use only. Customer will receive message based exclusively on the contents of the *Text* field.

If you have configured SMS templates of type Create, and you have them assigned to currently chosen *Queue*, you will have the option of populating the *Text* field with preconfigured message.

Flash message checkbox controls if message will be showed directly on the device without user interaction and not stored automatically.

Note

Support for flash messages (also known as *Class 0* messages) may vary by device and provider. OTRS cannot guarantee if this flag will be honored by service provider or supported by receiving device.

By clicking *Send SMS* button, ticket will be created with an SMS article, provided SMS cloud service accepted the message. If any errors were encountered, you will be notified, and ticket will not be created.

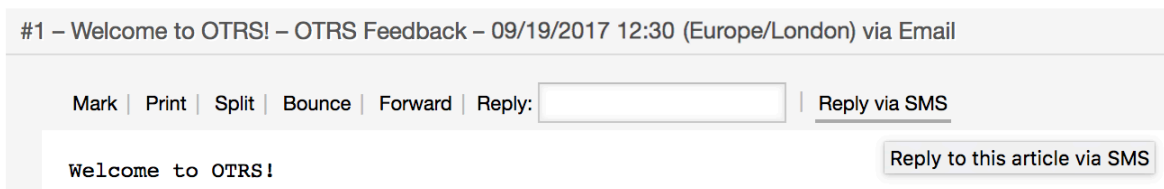
11.5. Creating SMS Articles

To add an SMS to an existing ticket, you can use *SMS Outbound* action on the main ticket toolbar (*Communication* section).

Figure 3.61. New SMS Ticket Screen

If you already have an article, you can also use *Reply via SMS* action on the article toolbar. Simply select the article in question, and either click on the button, or choose name of the template (if you have it configured for this screen).

Figure 3.62. New SMS Ticket Screen



If customer user has defined phone number, it will be automatically inserted in the recipients list. It doesn't matter if article was created in a different communication channel (e.g. Email), it should be correctly recognized and substituted with valid phone number.

Note

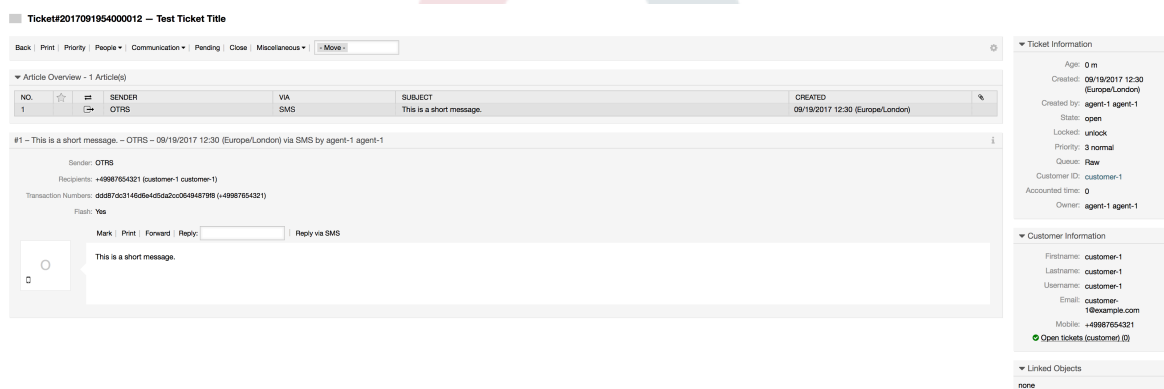
Please note that SMS character counter might not be reflecting the final message size. This is especially true if message text contains OTRS tags or links and URL shorting feature is turned on. Since text replacement will occur during sending of the message, there is no way to guarantee final size, and counter should be taken only for informative purposes.

11.6. Displaying SMS Articles

SMS articles can be displayed as they were sent out right in the ticket zoom screen. Their contents will reflect the final state of the message as queued by the SMS cloud service for sending (with all tags replaced and links shortened).

Recipients of an SMS article are listed inside the *Recipients* field. They can be shown in two formats: just phone numbers or phone numbers with name of corresponding customer users. In addition to this field, *Transaction Numbers* will contain a reference message ID mapped to the recipients phone number. These numbers can be used for debugging purposes, since they can be connected to specific messages handled by the cloud service.

Figure 3.63. SMS Article Display



In addition to the article header field, transaction numbers can also be found in *Ticket History* screen. This screen will contain an entry for every message sent out via SMS cloud service, with corresponding transaction and phone number in it.

Figure 3.64. SMS Ticket History Entries

AGENT	ACTION	ARTICLE
agent-1 agent-1	Created ticket [2017091954000012] in "Raw" with priority "3 normal" and state "open".	
agent-1 agent-1	Changed customer to "CustomerID=customer-1;CustomerUser=customer-1;".	
agent-1 agent-1	Added SMS for "+49987654321".	🔍
agent-1 agent-1	Sent SMS to +49987654321 (ddd87dc3146d6e4d5da2cc06494879f8).	
agent-1 agent-1	Reset of unlock time.	
agent-1 agent-1	Changed owner to "agent-1" (2).	

12. The Article Attachment Overview

12.1. Description

The article attachment overview enables you to view a list of all attachments of a ticket. Attachments will be listed by attributes such as file name, file type, file size, date of addition and which article they belong to. From the overview screen, you can download certain attachments or create an archive file which contains multiple or all attachments of the ticket you're viewing.

12.2. Configuration

No further configuration needed.

12.3. Usage

Figure 3.65. Switching to the article attachment overview

Back | Print | Priority | Attachments | People ▼ | Communication ▼

Once a ticket has at least one attachment, the *Attachments* menu item will be displayed in the ticket zoom screen. Clicking it will open a popup which contains a list of all attachments.

Figure 3.66. Using the article attachment overview

Attachments of Ticket#: 2017101654000015 - Test Ticket
 Close

Attachment Overview Hide inline attachments

Filter attachments...

<input type="checkbox"/>	TYPE	FILENAME	SIZE	▼ DATE	⇄	ARTICLE
<input type="checkbox"/>		StdAttachment-Test1.png	19.6 KB	16/10/2017 10:04 (Europe/London)	⇄	#3 - Note
<input type="checkbox"/>		StdAttachment-Test1.xls	8.5 KB	16/10/2017 10:04 (Europe/London)	⇄	#3 - Note
<input type="checkbox"/>		StdAttachment-Test1.pdf	4.6 KB	16/10/2017 10:04 (Europe/London)	→	#2 - Re: [[Ticket#2017101654000015] Tes...
<input type="checkbox"/>		StdAttachment-Test1.txt	392 B	16/10/2017 10:04 (Europe/London)	→	#2 - Re: [[Ticket#2017101654000015] Tes...
<input type="checkbox"/>		StdAttachment-Test1.doc	28.5 KB	16/10/2017 10:03 (Europe/London)	←	#1 - Test Ticket

Close

13. The Ticket Timeline View

13.1. Description

The ticket timeline view provides a chronological view of all actions which happened on a ticket (which includes articles, owner changes, incoming mails, etc.). With the ticket timeline view, agents are able to get a good overview of a ticket in less time.

13.2. Configuration

System Configuration

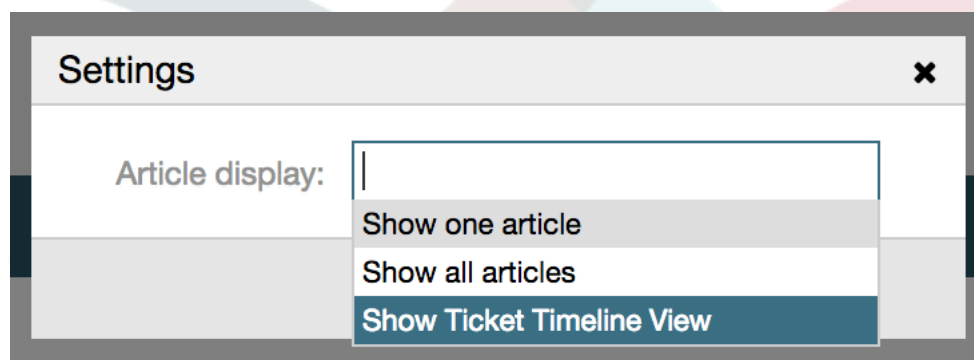
ChronicalViewEnabled

Group: OTRSBusiness, Subgroup: Frontend::Agent

Controls whether or not the ticket timeline view is enabled.

13.3. Usage

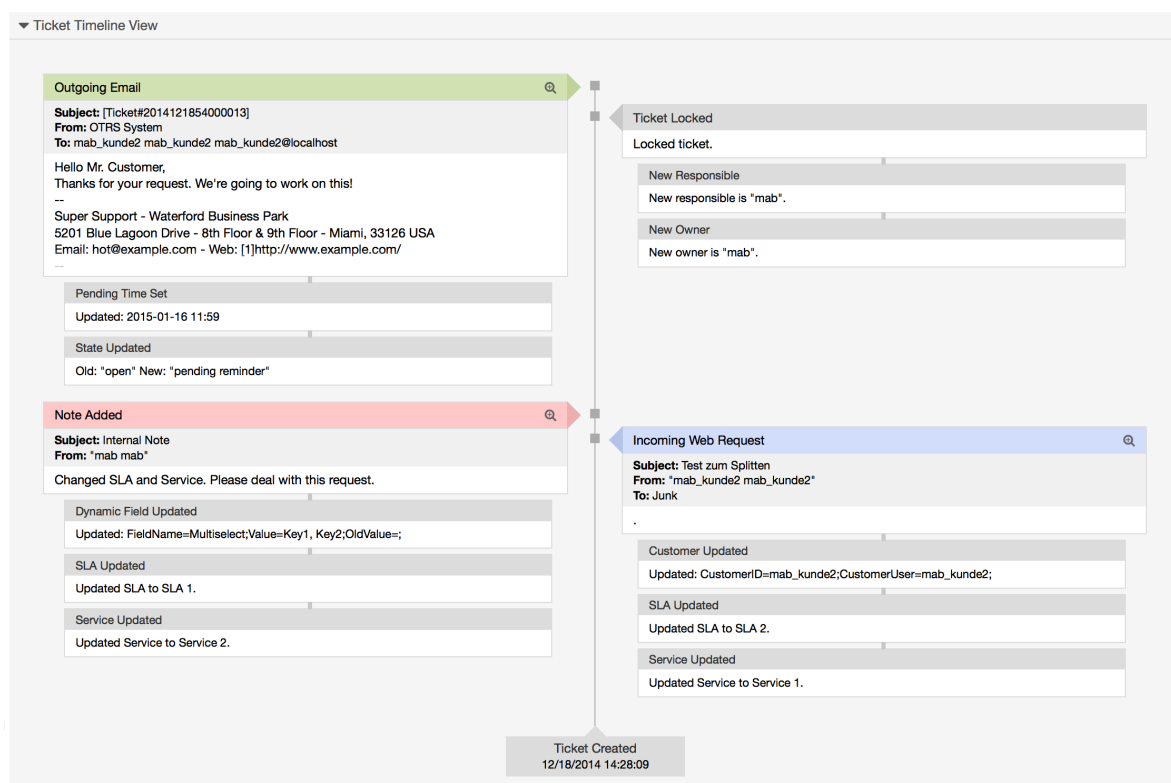
Figure 3.67. Switching to the Ticket Timeline View



To access the ticket timeline view, use the clock icon on the top right corner of the article widget in the ticket zoom screen. You will be redirected to the timeline view and this view will be kept as preference until you change back to another view.

In the ticket timeline view, you can work on the ticket as you're used to. You can inspect the time of each event in the timeline by hovering the little square next to each event with your mouse cursor. You can view the full content of articles by using the zoom icon on top of the certain articles box, where you will find all common article actions such as reply, split, forward, etc.

Figure 3.68. Using the Ticket Timeline View



14. System Configuration User Specific Settings

14.1. Description

This functionality allows agent users to define their own values for some system configuration settings. These values can be defined in the user preferences section in the new *Advanced* pane.

Administrators are able to review the user defined values for those settings and reset them if necessary and to forbid or allow user modifications on specific settings.

14.2. Configuration

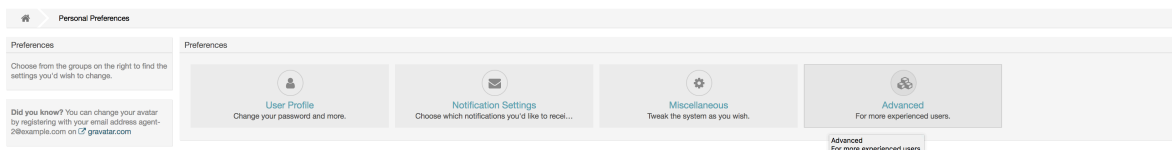
No further configuration needed.

14.3. Usage

14.3.1. Set user specific value

Every agent can define a value for a setting using the *Personal preferences* screen. Once you are in this screen, select the *Advanced* option in the preferences panel.

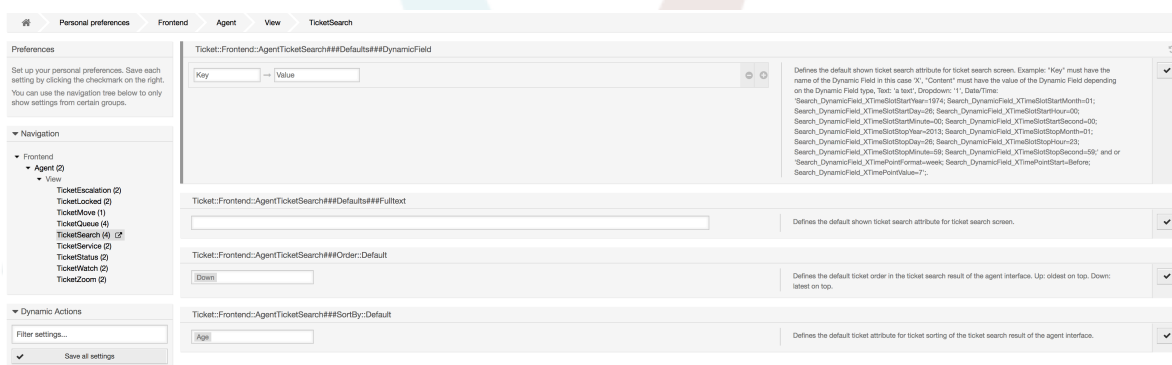
Figure 3.69. Accessing the user specific settings



Using the navigation widget in the left side bar, select the desired group of settings. Then you can change the value for a specific setting and save it using the *check mark* button in the right side of the setting widget.

Once you modified a value, it is possible to revert this change using the *Reset to default* icon button present in the right top corner of the modified setting widget.

Figure 3.70. Reverting a setting to default value

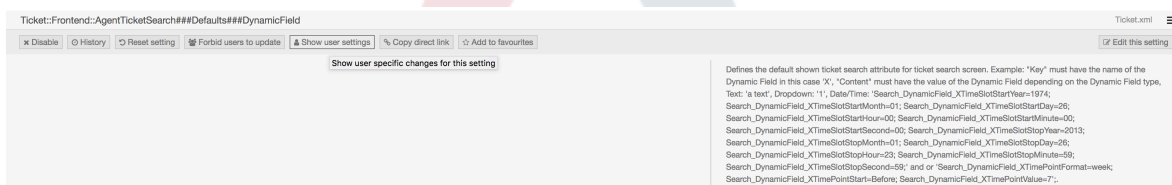


14.3.2. Review user specific settings

When a setting is available for user modification, all users are able to set their own values. Depending on the nature of the setting, values can be very different among users, and also compared to the original global value. This package includes an administrative tool to review and delete user set values if needed.

Click on the *Show user settings* button from the alternative menu on the settings that are enabled to show the *Review users setting values* screen.

Figure 3.71. Access to user values of a particular setting



On this screen all of the user specific values for the selected setting will be shown. To delete a single value click on the trash can icon in the right corner of every setting widget, or delete all user values for this setting by clicking on the trash icon in the outer setting widget.

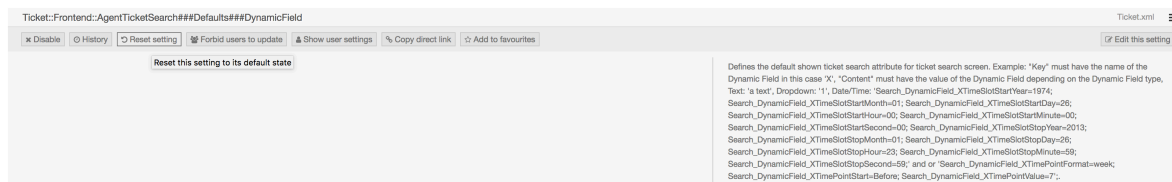
Figure 3.72. Usage of the user specific values screen



14.3.3. Reset globally and locally

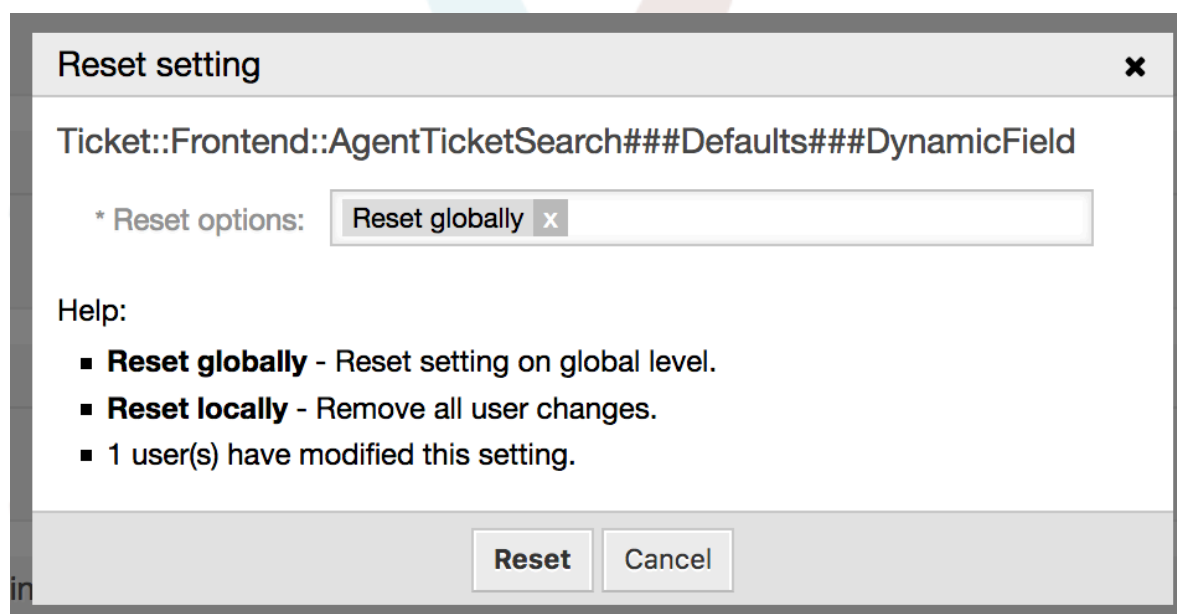
For the settings that are enabled to hold user values *Reset setting* feature has been enhanced to either reset the setting completely or just the user values.

Figure 3.73. Access to the setting reset dialog



Clicking on the *Reset setting* button will display a dialog showing the possible reset options.

Figure 3.74. Choosing the reset scope



15. System Configuration History

15.1. Description

This added functionality provides administrators with screens to go through system configuration deployment and specific settings history, and make changes if needed.

15.2. Configuration

No further configuration needed.

15.3. Usage

15.3.1. Deployment History

To access deployment history, go to *System Configuration* → *Deployment* screen and click on *History* button. This screen will display recent deployments in a descending order, along with their comments and name of the user who deployed the changes.

Figure 3.75. System configuration deployment history

System configuration

System Configuration > Changes Deployment > Deployment History

Actions	Recent Deployments	COMMENT	RESPONSIBLE	DETAILS	RESTORE	EXPORT
Go back	16/10/2017 09:03 (Europe/London)	Revert	agent-1 agent-1	View Details		⬇
Filter for Deployments	16/10/2017 09:02 (Europe/London)	FQDN change	agent-1 agent-1	View Details	⌵	⬇
Just start typing to filter...	16/10/2017 08:47 (Europe/London)	Configuration Rebuild	Admin OTRS	View Details	⌵	⬇
	16/10/2017 08:46 (Europe/London)	Secure mode	Admin OTRS	View Details	⌵	⬇
	16/10/2017 08:46 (Europe/London)	Configuration Rebuild	Admin OTRS	View Details	⌵	⬇

Every deployment can be further inspected by clicking on *View Details* link next to it. *Details* screen can be used to compare the setting with its previous value, before the deployment took place.

Additionally, older deployments (every one before current state) can be restored with a simple click. By restoring a deployment, all settings will be reverted to the value they had at the time of the deployment in question.

Finally, deployments can be exported with click on the export button. User will be presented with a download of a YML file that contains changed settings. This YML file can be later restored via *Import & Export* screen in the system configuration screen.

15.3.2. Setting History

Specific setting history can be accessed via *History* button in the setting widget. This button opens a screen of all values setting had over different deployments. Information like name of the user that made the change and time of change is displayed, along with a useful comparison tool.

Figure 3.76. System configuration setting history

System configuration

System Configuration > Changes Deployment > Deployment History

Actions	Recent Deployments	COMMENT	RESPONSIBLE	DETAILS	RESTORE	EXPORT
Go back	16/10/2017 09:03 (Europe/London)	Revert	agent-1 agent-1	View Details		⬇
Filter for Deployments	16/10/2017 09:02 (Europe/London)	FQDN change	agent-1 agent-1	View Details	⌵	⬇
Just start typing to filter...	16/10/2017 08:47 (Europe/London)	Configuration Rebuild	Admin OTRS	View Details	⌵	⬇
	16/10/2017 08:46 (Europe/London)	Secure mode	Admin OTRS	View Details	⌵	⬇
	16/10/2017 08:46 (Europe/London)	Configuration Rebuild	Admin OTRS	View Details	⌵	⬇

Every historical setting value can be restored by clicking on the *Restore* button in upper right corner.

Appendix A. Additional Resources

otrs.com

The OTRS website with source code, documentation and news is available at www.otrs.com. Here you can also find information about professional services and OTRS Administrator training seminars from OTRS Group, the creator of OTRS.

Translations

You can help translate OTRS to your language at Transifex.

