OTRS Administration Manual

Release 8.0

OTRS AG

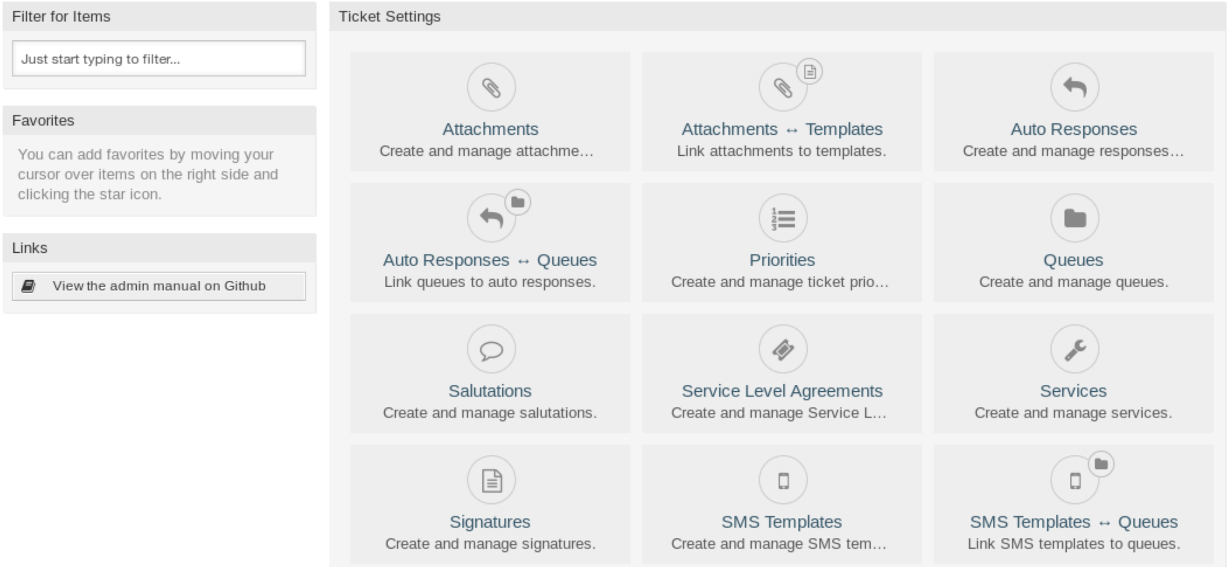
Jan 20, 2021

Contents

CHAPTER 1

Introduction

This manual is intended for use by OTRS administrators. The chapters describe the administration of the OTRS software. The administrator interface is available in the *Admin* menu item of the main menu, when you logged in as an administrator. Administrators are users, who are member of the *admin* group.



Filter for Items

Just start typing to filter...

Favorites

You can add favorites by moving your cursor over items on the right side and clicking the star icon.

Links

View the admin manual on Github

Ticket Settings

Attachments
Create and manage attachme…

Attachments ↔ Templates
Link attachments to templates.

Auto Responses
Create and manage responses…

Auto Responses ↔ Queues
Link queues to auto responses.

Priorities
Create and manage ticket prio…

Queues
Create and manage queues.

Salutations
Create and manage salutations.

Service Level Agreements
Create and manage Service L…

Services
Create and manage services.

Signatures
Create and manage signatures.

SMS Templates
Create and manage SMS tem…

SMS Templates ↔ Queues
Link SMS templates to queues.

Fig. 1: Administrator Interface

The administrator interface contains several modules collected into groups. Use the filter box in the left sidebar to find a particular module by just typing the name to filter.

This manual shows you the configuration possibilities needed to solve common problems. The chapters:

1. Identify a typical use-case for the administrator, to aid in orientation, and explain **what** OTRS does to provide a solution (**warranty**).
2. Direct you **how** to configure OTRS to fit your use-case (**utility**).

The chapters are the same as the modules in the administrator interface. The order of the chapters are also the same as they are displayed alphabetically in the (English) administrator interface. However, the steps to do to configure a fresh new OTRS installation is different from this order. We recommend to configure OTRS to do the steps as explained below.

1.1 Getting Started

Note: **OTRS** is installed by the *Customer Solution Team*. Please contact the *Customer Solution Team* via support@otrs.com or in the OTRS Portal.

A fresh new OTRS installation contains only the default settings. You need to customize your system to work properly and meet your needs.

First, you need to check some *System Configuration* and modify the values, if needed. FQDN, SystemID and SendmailModule::Host are the most important. Defining the TimeWorkingHours setting for working hours and the TimeVacationDays setting for public holidays are also needed to calculate the escalation times correctly in OTRS.

Then, open the *PostMaster Mail Accounts* module to add email addresses used by the system. For sending email, you can add more *Email Addresses*.

To improve the security, *PGP Keys* or *S/MIME Certificates* can be used for encryption of emails.

Let’s continue with users, but you might need to add some *Groups* and/or *Roles* to the system first. It is recommended to create new groups for each main queues. There are some groups in your OTRS, but no roles are defined by default. You can add roles, if needed, and you can set the *Roles Groups* relations.

The next step is to add *Agents* to the system and set the *Agents Groups* and/or *Agents Roles* relations.

Now you can add *Customers* and *Customer Users*. Customers are companies and customer users are the employees of the company.

Note: Both agents and customer users can log in using Active Directory or LDAP for authentication. In these cases doesn’t need to add users manually.

Like for agents, customers users can also assign to groups using the *Customer Users Groups* management screen. Similarly, *Customer Users Customers* relations can also be set.

Your OTRS installation already contains a standard salutation and a standard signature, but you might need to edit them or create new *Salutations* and *Signatures*. Queues can not be created without salutations and signatures, and only one salutation and signature can be assigned to a queue.

After system addresses, salutations, signatures, groups are set, you can continue the setup with *Queues*. Each queue has to assign to a group, and only the group members can see the tickets in the assigned queue.

Now you can add *Auto Responses* and assign them to queues using the *Queues Auto Responses* management screen. Your OTRS installation already contains some automatic responses, you can use or edit them instead of create new ones.

To reduce the time needed for answering the tickets, *Templates* or *SMS Templates* can be created.

Normal templates can contain *Attachments*, and you can assign the uploaded attachments to templates using the *Templates Attachments* management screen.

When templates are created, you can set the templates to use in queues in the *Templates Queues* or *SMS Templates Queues* management screens.

You need to review the default *Priorities*, *States* and *Types*, and add new elements, if needed.

The customer requests can be categorize into services. If you would like to use this possibility, then create some *Services* and set the *Customer Users Services* relations. Furthermore, *Service Level Agreements* can be assign to the services.

Now you can add some notifications to be received by agents, if particular events occur. You can do this in the *Ticket Notifications* screen.

To help agents to organize appointments, you can setup the *Calendars* and the *Appointment Notifications*.

Tickets, articles and other objects in OTRS can be extended with *Dynamic Fields* or can be reduced with *Access Control Lists (ACL)*.

Without doing everything manually, the number of failure can be reduced. Automatize some process in OTRS using *Generic Agent* jobs or creating processes with *Process Management*. The incoming emails can be pre-processed and dispatched automatically by defining some *Postmaster Filters*.

If external systems need to integrate with OTRS, *Web Services* will be very useful for this.

However OTRS has many features by default, you can extend the functionality by installing packages with the *Package Manager*.

If your system is ready for productive work, don’ t forget to register it by using the *System Registration* procedure.

Finally, you can set the *Home Page*, the *Custom Pages* and the *Layout* of the external interface, as well as you can define a *Customer Service Catalogue* displayed in the external interface.

1.2 Become OTRS Expert

The next chapters of this manual describe the features and configuration settings of OTRS more detailed. There is a separated manual for Configuration Options References, that gives you a good overview of *System Configuration*, that can be modify the behavior of OTRS.

CHAPTER 2

Ticket Settings

A *ticket* is similar to a medical report created for a hospital patient. When a patient first visits the hospital, a medical report is created to hold all necessary personal and medical information on him. Over multiple visits, as he is attended to by the same or additional doctors, the attending doctor updates the report by adding new information on the patient’s health and the ongoing treatment. This allows any other doctors or the nursing staff to get a complete picture on the case at hand. When the patient recovers and leaves the hospital, all information from the medical report is archived and the report is closed.

Ticket systems such as OTRS handle tickets like normal emails. The messages are saved in the system. When a customer sends a request, a new ticket is generated by the system which is comparable to a new medical report being created. The response to this new ticket is comparable to a doctor’s entry in the medical report. A ticket is closed if an answer is sent back to the customer, or if the ticket is separately closed by the system. If a customer responds again on an already closed ticket, the ticket is reopened with the new information added.

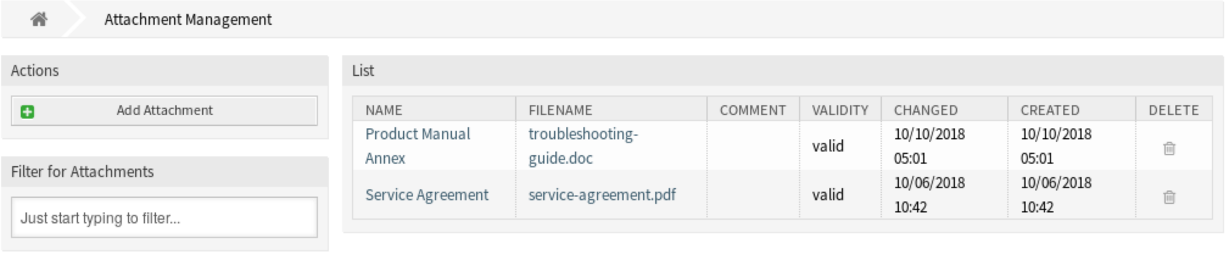
Every ticket is stored and archived with complete information. Since tickets are handled like normal emails, attachments and contextual annotations will also be stored with each email. In addition, information on relevant dates, employees involved, working time needed for ticket resolution, etc. are also saved. At any later stage, tickets can be sorted, and it is possible to search through and analyze all information using different filtering mechanisms.

2.1 Attachments

For any size of organization it is often required to send a service agreement, the terms of service or a privacy statement out when a customer signs a contract.

OTRS can handle an infinite number of attachments (PDF, image, etc.) and can bundle them into templates. Your agents don’t need to maintain the attachments on their own, nor don’t they need to upload them again and again - they can just use the predefined templates.

Use this screen to add attachments for use in templates. A fresh OTRS installation doesn’t contain any attachments by default. The attachment management screen is available in the *Attachments* module of the *Ticket Settings* group.



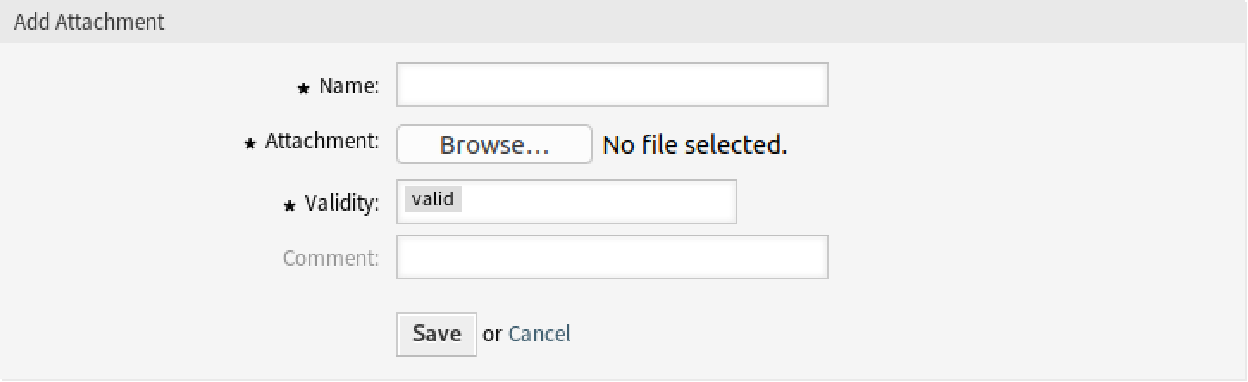
NAME	FILENAME	COMMENT	VALIDITY	CHANGED	CREATED	DELETE
Product Manual Annex	troubleshooting-guide.doc		valid	10/10/2018 05:01	10/10/2018 05:01	🗑
Service Agreement	service-agreement.pdf		valid	10/06/2018 10:42	10/06/2018 10:42	🗑

Fig. 1: Attachment Management Screen

2.1.1 Manage Attachments

To add an attachment:

1. Click on the *Add Attachments* button in the left sidebar.
2. Fill in the required fields.
3. Click on the *Save* button.



⋆ Name:	
⋆ Attachment:	No file selected.
⋆ Validity:	valid
Comment:	

Fig. 2: Add Attachment Screen

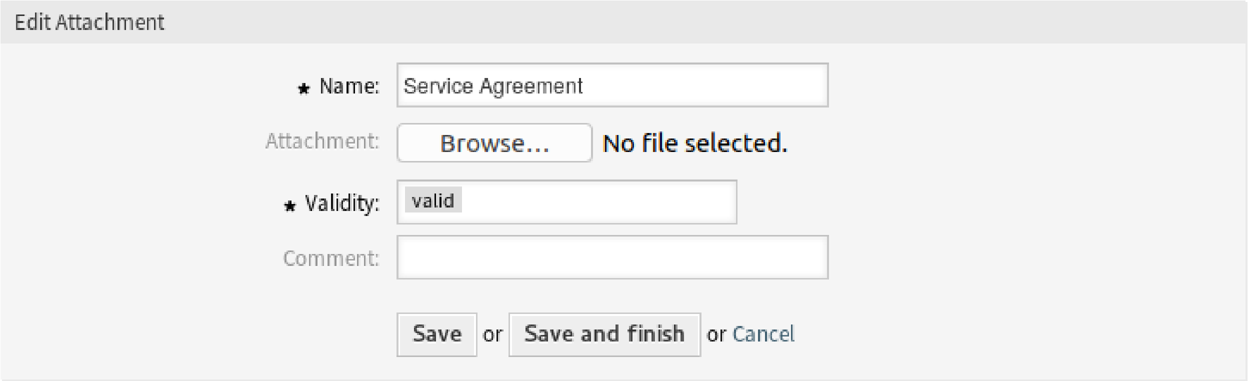
To edit an attachment:

1. Click on an attachment in the list of attachments.
2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.

To delete an attachment:

1. Click on the trash icon in the last column of the overview table.
2. Click on the *Confirm* button.

Note: If several attachments are added to the system, use the filter box to find a particular attachment by just typing the name to filter.


Edit Attachment

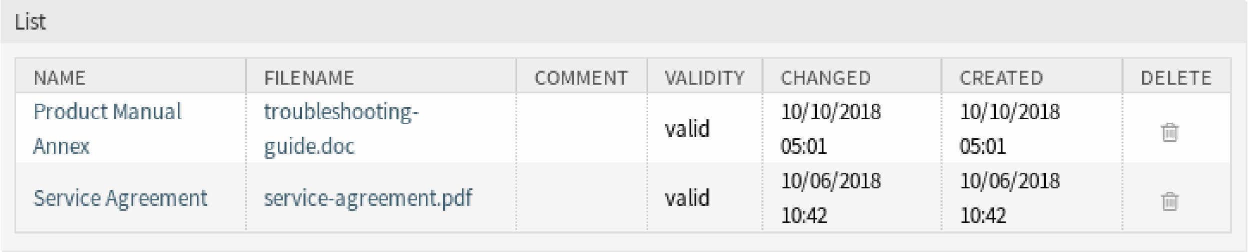
⋆ Name: Service Agreement

Attachment: No file selected.

⋆ Validity: valid

Comment:

Fig. 3: Edit Attachment Screen


List

NAME	FILENAME	COMMENT	VALIDITY	CHANGED	CREATED	DELETE
Product Manual Annex	troubleshooting-guide.doc		valid	10/10/2018 05:01	10/10/2018 05:01	🗑
Service Agreement	service-agreement.pdf		valid	10/06/2018 10:42	10/06/2018 10:42	🗑

Fig. 4: Delete Attachment Screen

2.1.2 Attachment Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Attachment * Open the file dialog to add a file from the file system. This field is mandatory in the attachment add screen, but optional in the attachment edit screen. Adding a new file in the edit screen will overwrite the existing attachment.

Validity * Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

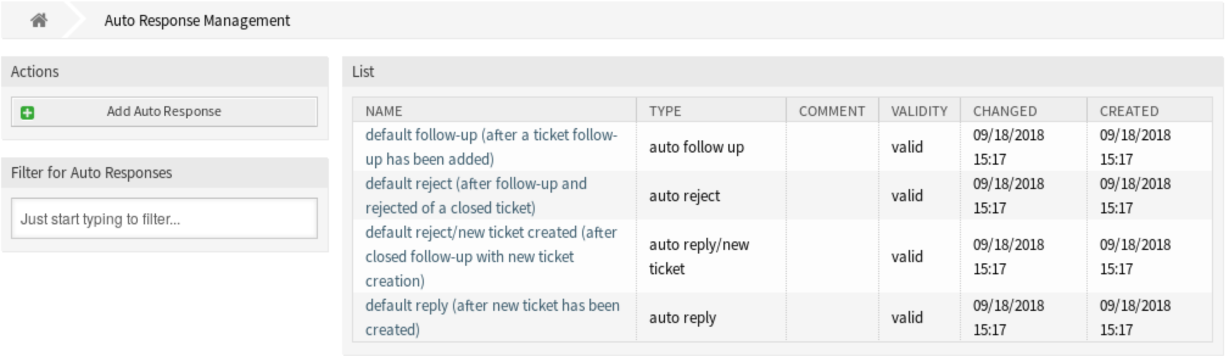
2.2 Auto Responses

Quick and transparent service is vital to maintaining a good working relationship with your customer. Email, fax, social media and other non-real-time communication are patient, but you want to engage your customer immediately upon receipt of a request.

OTRS allows you to respond to a customer immediately upon receipt of a request giving the customers instantaneous feedback assuring them that their request is in processing, establishing expectation.

Automatic responses can be sent to customers based on the occurrence of certain events, such as the creation of a ticket in a specific queue, the receipt of a follow-up message in regards to a closed or rejected ticket, etc.

Use this screen to add automatic responses for use in queues. A fresh OTRS installation contains some automatic responses by default. The automatic response management screen is available in the *Auto Responses* module of the *Ticket Settings* group.



NAME	TYPE	COMMENT	VALIDITY	CHANGED	CREATED
default follow-up (after a ticket follow-up has been added)	auto follow up		valid	09/18/2018 15:17	09/18/2018 15:17
default reject (after follow-up and rejected of a closed ticket)	auto reject		valid	09/18/2018 15:17	09/18/2018 15:17
default reject/new ticket created (after closed follow-up with new ticket creation)	auto reply/new ticket		valid	09/18/2018 15:17	09/18/2018 15:17
default reply (after new ticket has been created)	auto reply		valid	09/18/2018 15:17	09/18/2018 15:17

Fig. 5: Auto Response Management Screen

2.2.1 Manage Auto Responses

Note: Adding automatic responses requires a valid system address. Create system addresses in the *Email Addresses* screen.

To add an automatic response:

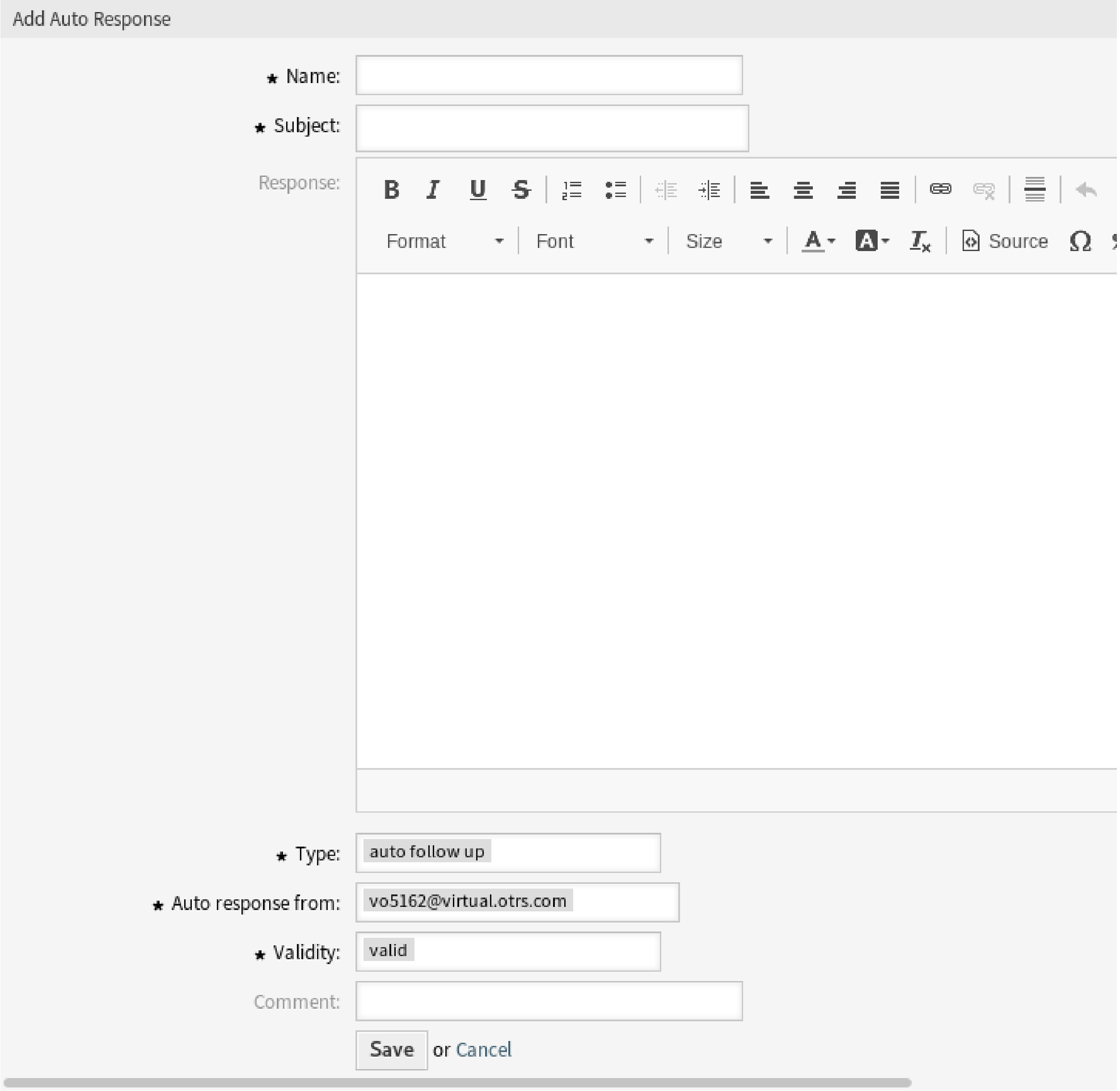
1. Click on the *Add Auto Response* button in the left sidebar.
2. Fill in the required fields.
3. Click on the *Save* button.

Warning: Auto responses can not be deleted from the system. They can only be deactivated by setting the *Validity* option to *invalid* or *invalid-temporarily*.

To edit an automatic response:

1. Click on an automatic response in the list of automatic responses.
2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.

Note: If several automatic responses are added to the system, use the filter box to find a particular automatic response by just typing the name to filter.



Add Auto Response

★ Name:

★ Subject:

Response:

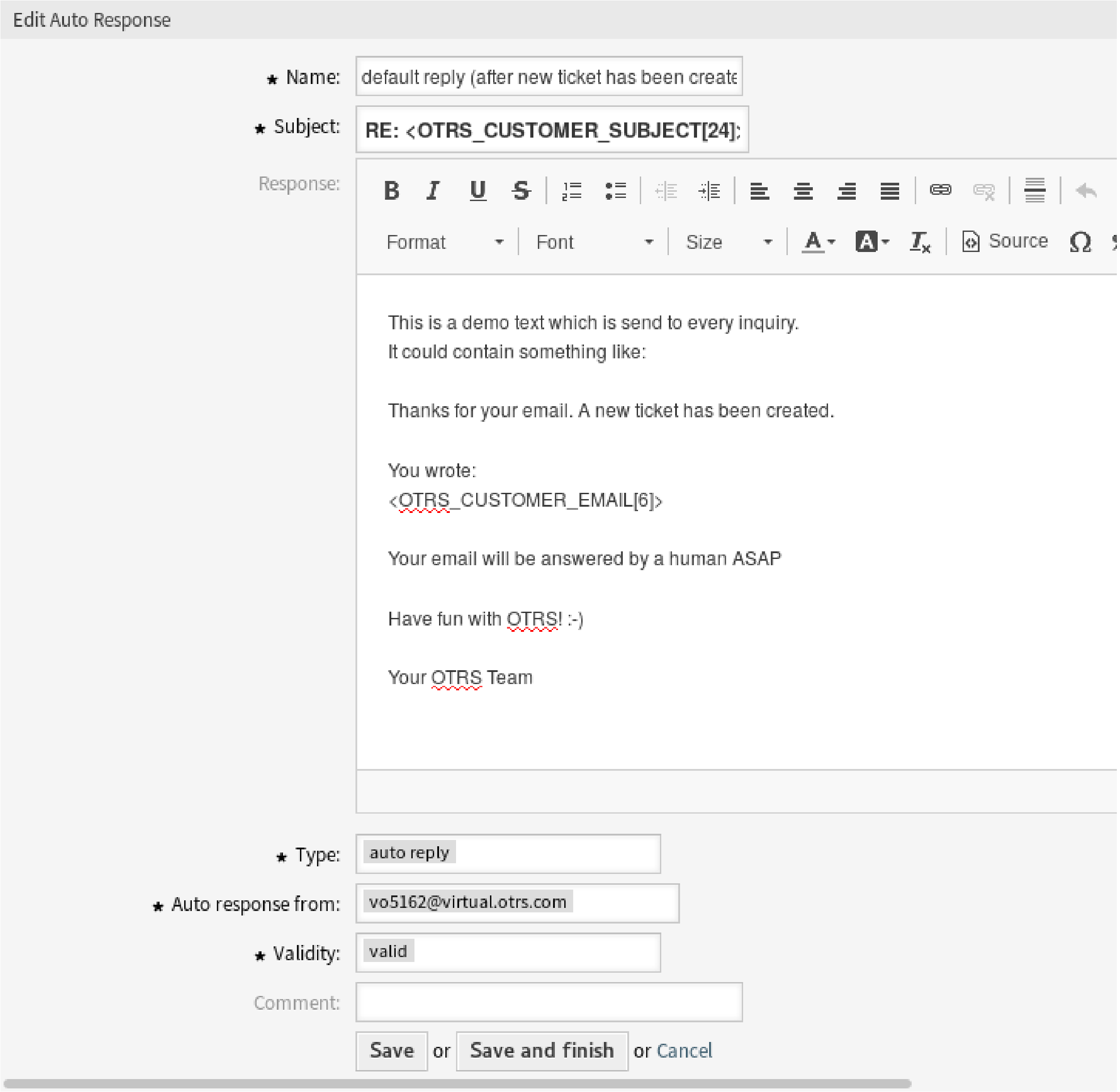
★ Type: auto follow up

★ Auto response from: vo5162@virtual.otrs.com

★ Validity: valid

Comment:

Fig. 6: Add Auto Response Screen



Edit Auto Response

★ Name: default reply (after new ticket has been create

★ Subject: RE: <OTRS_CUSTOMER_SUBJECT[24]>

Response:

This is a demo text which is send to every inquiry.
It could contain something like:

Thanks for your email. A new ticket has been created.

You wrote:
<OTRS_CUSTOMER_EMAIL[6]>

Your email will be answered by a human ASAP

Have fun with OTRS! :-)

Your OTRS Team

★ Type: auto reply

★ Auto response from: vo5162@virtual.otrs.com

★ Validity: valid

Comment:

Fig. 7: Edit Auto Response Screen

2.2.2 Auto Response Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Subject * The subject of the email sent to the users.

Response The body of the email sent to the users.

Type * The event type that triggers the sending of this automatic response. Only one automatic response can be sent automatically. The following event types are available:

- **auto follow up** Confirms receipt of the follow-up.
- **auto reject** The message rejects a customer follow-up.
- **auto remove** Deletion of a ticket, done by the system.
- **auto reply** A newly raised ticket will trigger this auto response.
- **auto reply/new ticket** This message informs the customer of the new ticket number.

Auto response from * The sender email address, from which the automatic response will be sent.

Validity * Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

2.2.3 Auto Response Variables

Using variables in the text makes it possible to personalize messages. Variables, known as OTRS tags, are replaced by OTRS when generating the message. Find a list of available tags stems for this resource at the bottom of both add and edit screens.

For example, the variable `<OTRS_TICKET_TicketNumber>` expands to the ticket number allowing a template to include something like the following.

```
Ticket#<OTRS_TICKET_TicketNumber>
```

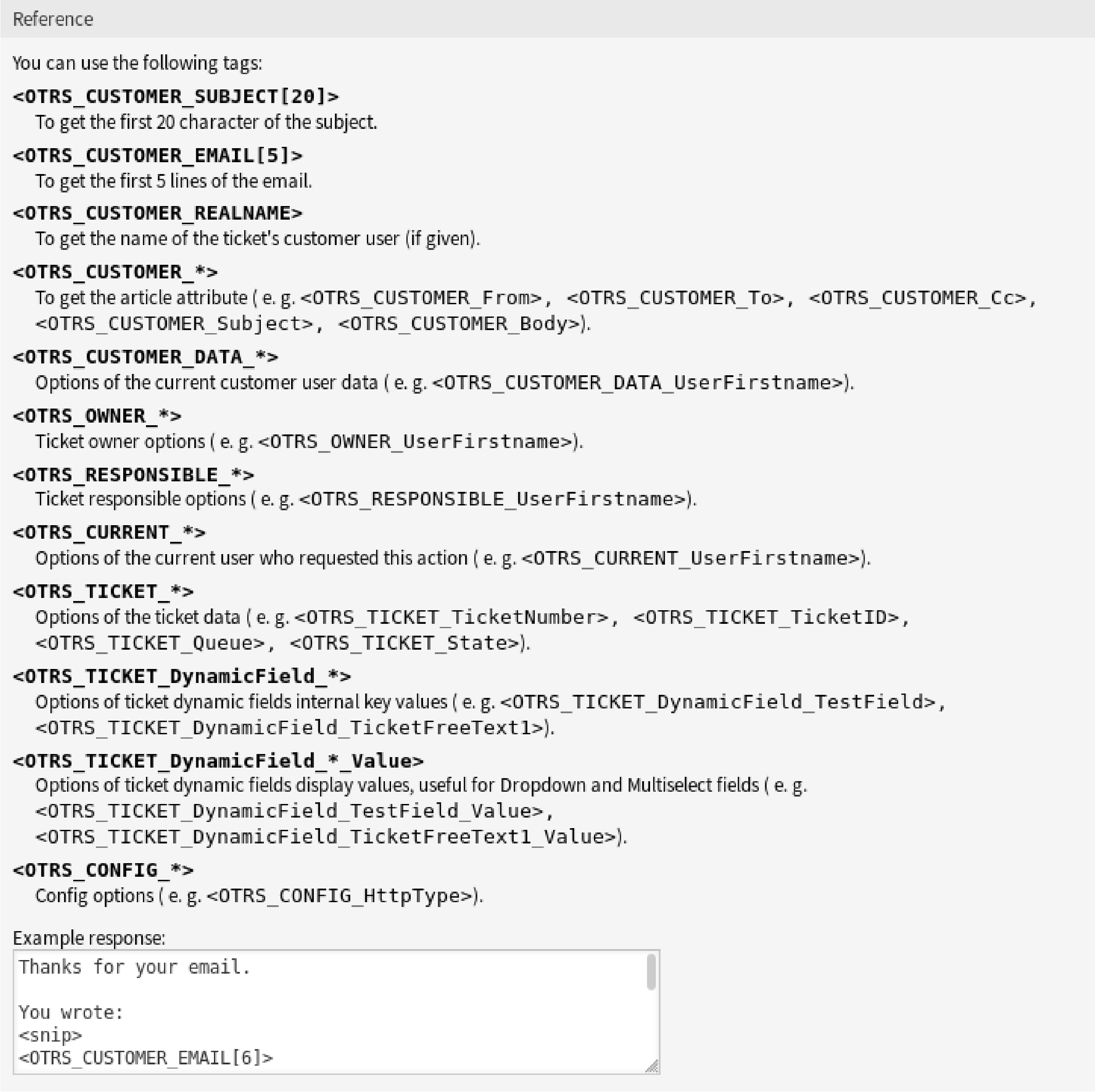
This tag expands, for example to:

```
Ticket#2018101042000012
```

The values of the following variables are translated based on the chosen language of the customer user.

```
OTRSTicket_Type
OTRSTicket_State
OTRSTicket_StateType
OTRSTicket_Lock
OTRSTicket_Priority
```

If the language is not supported the default language is applied.



Reference

You can use the following tags:

<OTRS_CUSTOMER_SUBJECT[20]>
To get the first 20 character of the subject.

<OTRS_CUSTOMER_EMAIL[5]>
To get the first 5 lines of the email.

<OTRS_CUSTOMER_REALNAME>
To get the name of the ticket's customer user (if given).

<OTRS_CUSTOMER_*>
To get the article attribute (e. g. `<OTRS_CUSTOMER_From>, <OTRS_CUSTOMER_To>, <OTRS_CUSTOMER_Cc>, <OTRS_CUSTOMER_Subject>, <OTRS_CUSTOMER_Body>`).

<OTRS_CUSTOMER_DATA_*>
Options of the current customer user data (e. g. `<OTRS_CUSTOMER_DATA_UserFirstname>`).

<OTRS_OWNER_*>
Ticket owner options (e. g. `<OTRS_OWNER_UserFirstname>`).

<OTRS_RESPONSIBLE_*>
Ticket responsible options (e. g. `<OTRS_RESPONSIBLE_UserFirstname>`).

<OTRS_CURRENT_*>
Options of the current user who requested this action (e. g. `<OTRS_CURRENT_UserFirstname>`).

<OTRS_TICKET_*>
Options of the ticket data (e. g. `<OTRS_TICKET_TicketNumber>, <OTRS_TICKET_TicketID>, <OTRS_TICKET_Queue>, <OTRS_TICKET_State>`).

<OTRS_TICKET_DynamicField_*>
Options of ticket dynamic fields internal key values (e. g. `<OTRS_TICKET_DynamicField_TestField>, <OTRS_TICKET_DynamicField_TicketFreeText1>`).

<OTRS_TICKET_DynamicField_*_Value>
Options of ticket dynamic fields display values, useful for Dropdown and Multiselect fields (e. g. `<OTRS_TICKET_DynamicField_TestField_Value>, <OTRS_TICKET_DynamicField_TicketFreeText1_Value>`).

<OTRS_CONFIG_*>
Config options (e. g. `<OTRS_CONFIG_HttpType>`).

Example response:

```
Thanks for your email.

You wrote:
<snip>
<OTRS_CUSTOMER_EMAIL[6]>
```

Fig. 8: Auto Response Variables

2.3 Priorities

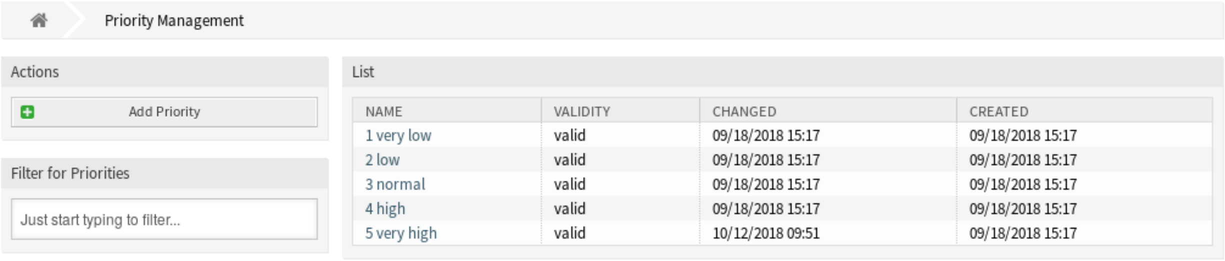
Sometimes tickets are not equally created. One ticket may need more focus than another. A customer may be given a higher priority by the service desk to help raise customer satisfaction in a pinch or to ensure that a long-running request receives special attention. Keeping track of these higher priority requests is important, as well as handling them quickly.

OTRS provides a traffic light system based per default five levels of priorities to handle this task.

- Blue: very low
- Green: low
- Grey: normal
- Pink: high
- Red: very high

The colors are based on the ID (very low being ID 1) in the database. Changing the names should be done with this in mind. New priorities have the color gray.

Use this screen to add priorities to the system. A fresh OTRS installation contains five default priority levels. The priority management screen is available in the *Priorities* module of the *Ticket Settings* group.



NAME	VALIDITY	CHANGED	CREATED
1 very low	valid	09/18/2018 15:17	09/18/2018 15:17
2 low	valid	09/18/2018 15:17	09/18/2018 15:17
3 normal	valid	09/18/2018 15:17	09/18/2018 15:17
4 high	valid	09/18/2018 15:17	09/18/2018 15:17
5 very high	valid	10/12/2018 09:51	09/18/2018 15:17

Fig. 9: Priority Management Screen

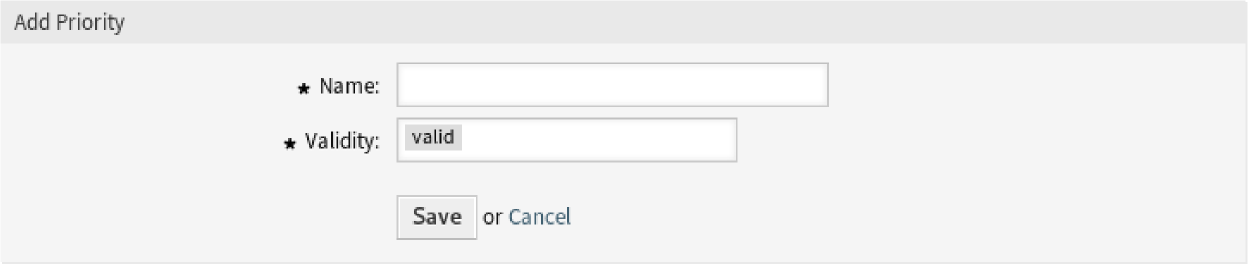
2.3.1 Manage Priorities

Note: When creating a customized list of priorities, please keep in mind that they are sorted alphabetically in the priority selection box in the user interface.

To add a priority:

1. Click on the *Add Priority* button in the left sidebar.
2. Fill in the required fields.
3. Click on the *Save* button.

Warning: Priorities can not be deleted from the system. They can only be deactivated by setting the *Validity* option to *invalid* or *invalid-temporarily*.



★ Name:

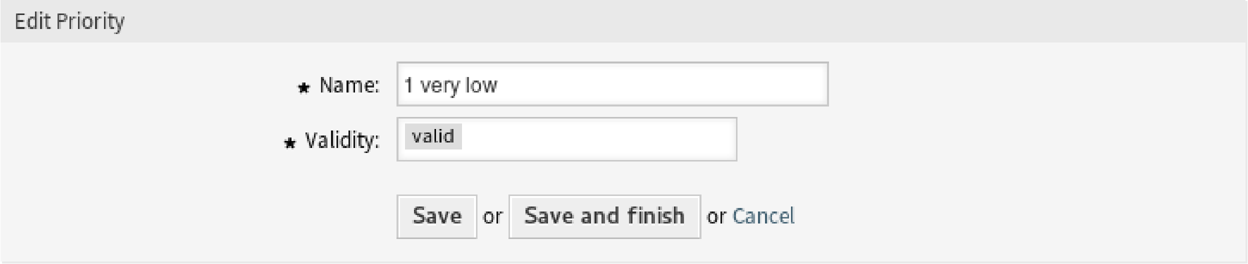
★ Validity: valid

Fig. 10: Add Priority Screen

Note: It's recommended to limit your system to 5 priorities or less and reuse the current 5 to keep the use of the traffic light system.

To edit a priority:

1. Click on a priority in the list of priorities.
2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.



★ Name: 1 very low

★ Validity: valid

Fig. 11: Edit Priority Screen

Note: If several priorities are added to the system, use the filter box to find a particular priority by just typing the name to filter.

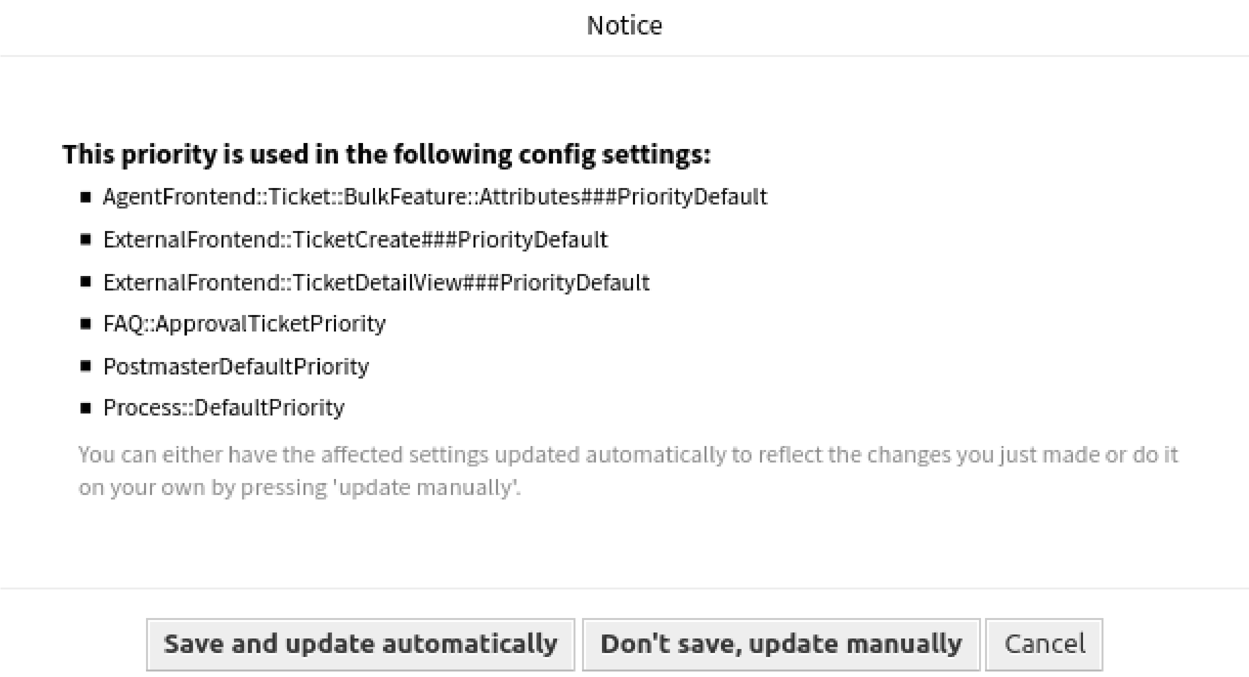
If you change the name of a priority which is used in the system configuration, a validation check will warn you and give you the following options:

Save and update automatically Apply the change and also update the affected settings.

Don't save, update manually Apply the change, but don't update the affected settings. The updates need to be done manually.

Cancel Cancel the action.

Warning: Changing the name of this object should be done with care, the check only provides verification for certain settings and ignores things where the name can't be verified. Some examples are dashboard filters, access control lists (ACLs), and processes (sequence flow actions) to name a few. Documentation of your setup is key to surviving a name change.



Notice

This priority is used in the following config settings:

- AgentFrontend::Ticket::BulkFeature::Attributes###PriorityDefault
- ExternalFrontend::TicketCreate###PriorityDefault
- ExternalFrontend::TicketDetailView###PriorityDefault
- FAQ::ApprovalTicketPriority
- PostmasterDefaultPriority
- Process::DefaultPriority

You can either have the affected settings updated automatically to reflect the changes you just made or do it on your own by pressing 'update manually'.

Fig. 12: Priority Validation Dialog

2.3.2 Priority Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Validity * Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.

2.4 Queues

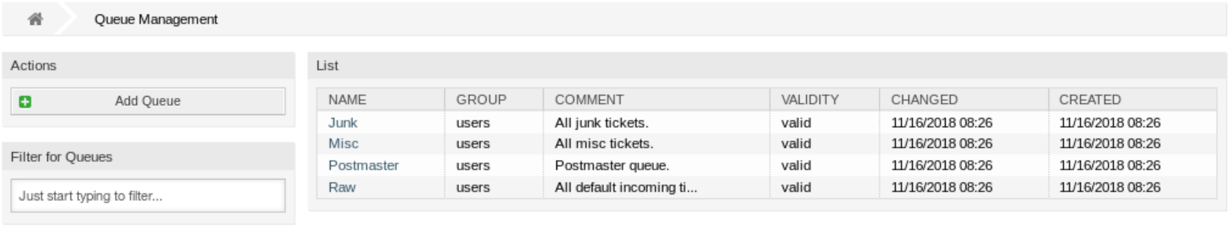
Teams need a workspace and the ability to dispatch work based on skill level, security level, department or responsibility just to name a few. Other teams may also need to view or react in these requests as well.

OTRS uses queues to provide your teams with structure. Queues provide a powerful way to divide and disperse the work to the responsible group of people.

Use this screen to add queues to the system. In a fresh OTRS installation there are 4 default queues: *Raw*, *Junk*, *Misc* and *Postmaster*. All incoming messages will be stored in the *Raw* queue if no filter rules are defined. The *Junk* queue can be used to store spam messages. The queue management screen is available in the *Queues* module of the *Ticket Settings* group.

2.4.1 Manage Queues

To add a queue:



NAME	GROUP	COMMENT	VALIDITY	CHANGED	CREATED
Junk	users	All junk tickets.	valid	11/16/2018 08:26	11/16/2018 08:26
Misc	users	All misc tickets.	valid	11/16/2018 08:26	11/16/2018 08:26
Postmaster	users	Postmaster queue.	valid	11/16/2018 08:26	11/16/2018 08:26
Raw	users	All default incoming ti...	valid	11/16/2018 08:26	11/16/2018 08:26

Fig. 13: Queue Management Screen

1. Click on the *Add Queue* button in the left sidebar.
2. Fill in the required fields.
3. Click on the *Save* button.

> **Warning:** Queues can not be deleted from the system. They can only be deactivated by setting the *Validity* option to *invalid* or *invalid-temporarily*.

To edit a queue:

1. Click on a queue in the list of queues.
2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.

Note: If several queues are added to the system, use the filter box to find a particular queue by just typing the name to filter.

If you change the name of a queue which is used in the system configuration, a validation check will warn you and give you the following options:

Save and update automatically Apply the change and also update the affected settings.

Don't save, update manually Apply the change, but don't update the affected settings. The updates need to be done manually.

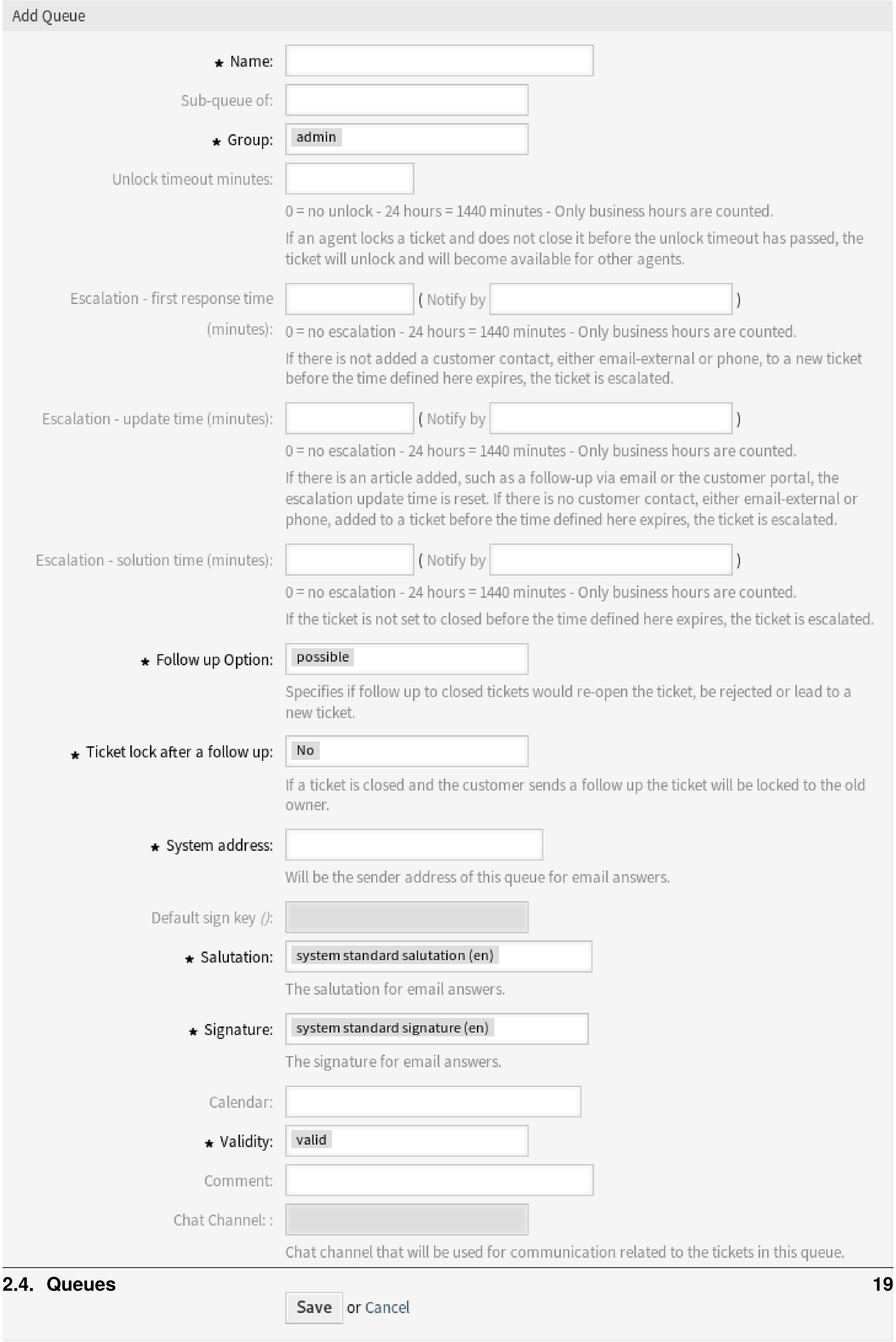
Cancel Cancel the action.

> **Warning:** Changing the name of this object should be done with care, the check only provides verification for certain settings and ignores things where the name can't be verified. Some examples are dashboard filters, access control lists (ACLs), and processes (sequence flow actions) to name a few. Documentation of your setup is key to surviving a name change.

2.4.2 Queue Settings

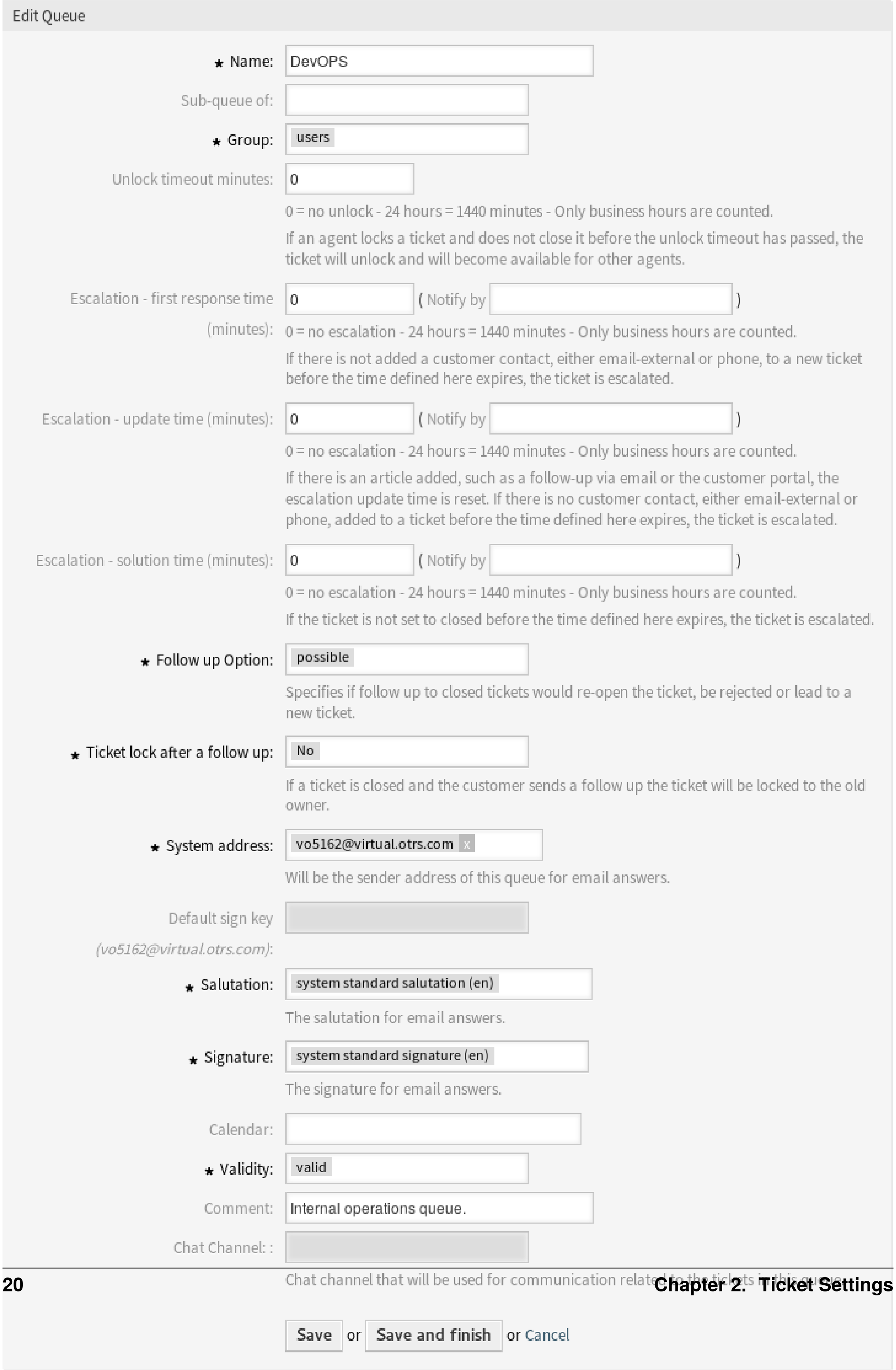
The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.



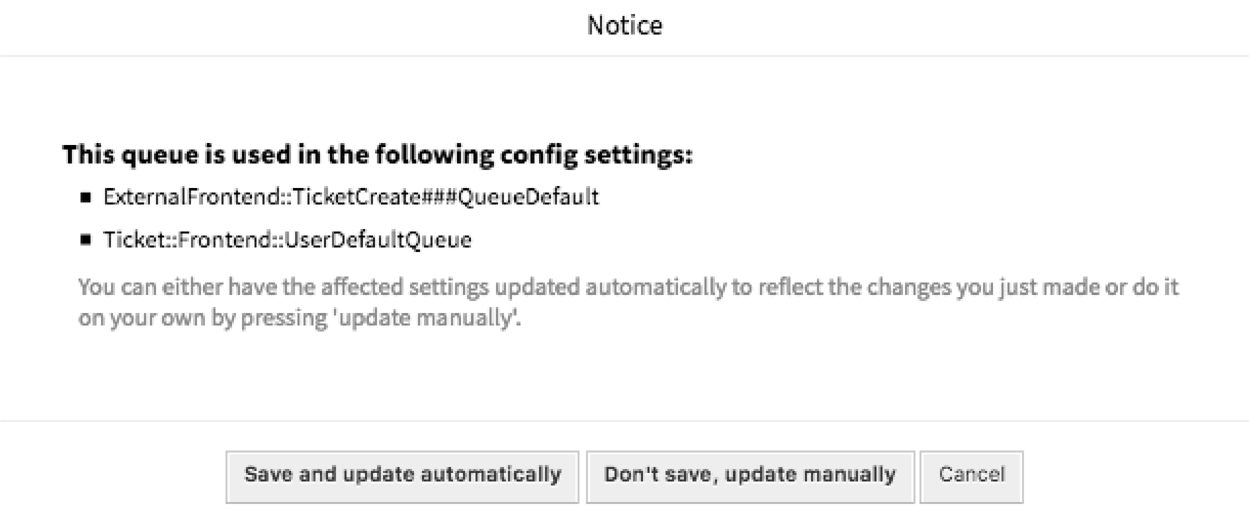
Add Queue

★ Name:	
Sub-queue of:	
★ Group:	admin
Unlock timeout minutes:	 0 = no unlock - 24 hours = 1440 minutes - Only business hours are counted. If an agent locks a ticket and does not close it before the unlock timeout has passed, the ticket will unlock and will become available for other agents.
Escalation - first response time (minutes):	(Notify by) 0 = no escalation - 24 hours = 1440 minutes - Only business hours are counted. If there is not added a customer contact, either email-external or phone, to a new ticket before the time defined here expires, the ticket is escalated.
Escalation - update time (minutes):	(Notify by) 0 = no escalation - 24 hours = 1440 minutes - Only business hours are counted. If there is an article added, such as a follow-up via email or the customer portal, the escalation update time is reset. If there is no customer contact, either email-external or phone, added to a ticket before the time defined here expires, the ticket is escalated.
Escalation - solution time (minutes):	(Notify by) 0 = no escalation - 24 hours = 1440 minutes - Only business hours are counted. If the ticket is not set to closed before the time defined here expires, the ticket is escalated.
★ Follow up Option:	possible Specifies if follow up to closed tickets would re-open the ticket, be rejected or lead to a new ticket.
★ Ticket lock after a follow up:	No If a ticket is closed and the customer sends a follow up the ticket will be locked to the old owner.
★ System address:	 Will be the sender address of this queue for email answers.
Default sign key ():	
★ Salutation:	system standard salutation (en) The salutation for email answers.
★ Signature:	system standard signature (en) The signature for email answers.
Calendar:	
★ Validity:	valid
Comment:	
Chat Channel: :	 Chat channel that will be used for communication related to the tickets in this queue.



Edit Queue

★ Name:	DevOPS
Sub-queue of:	
★ Group:	users
Unlock timeout minutes:	0 0 = no unlock - 24 hours = 1440 minutes - Only business hours are counted. If an agent locks a ticket and does not close it before the unlock timeout has passed, the ticket will unlock and will become available for other agents.
Escalation - first response time (minutes):	0 (Notify by) 0 = no escalation - 24 hours = 1440 minutes - Only business hours are counted. If there is not added a customer contact, either email-external or phone, to a new ticket before the time defined here expires, the ticket is escalated.
Escalation - update time (minutes):	0 (Notify by) 0 = no escalation - 24 hours = 1440 minutes - Only business hours are counted. If there is an article added, such as a follow-up via email or the customer portal, the escalation update time is reset. If there is no customer contact, either email-external or phone, added to a ticket before the time defined here expires, the ticket is escalated.
Escalation - solution time (minutes):	0 (Notify by) 0 = no escalation - 24 hours = 1440 minutes - Only business hours are counted. If the ticket is not set to closed before the time defined here expires, the ticket is escalated.
★ Follow up Option:	possible Specifies if follow up to closed tickets would re-open the ticket, be rejected or lead to a new ticket.
★ Ticket lock after a follow up:	No If a ticket is closed and the customer sends a follow up the ticket will be locked to the old owner.
★ System address:	vo5162@virtual.otrs.com x Will be the sender address of this queue for email answers.
Default sign key (vo5162@virtual.otrs.com):	
★ Salutation:	system standard salutation (en) The salutation for email answers.
★ Signature:	system standard signature (en) The signature for email answers.
Calendar:	
★ Validity:	valid
Comment:	Internal operations queue.
Chat Channel: :	 Chat channel that will be used for communication related to the tickets in this queue.



Notice

This queue is used in the following config settings:

- ExternalFrontend::TicketCreate###QueueDefault
- Ticket::Frontend::UserDefaultQueue

You can either have the affected settings updated automatically to reflect the changes you just made or do it on your own by pressing 'update manually'.

Fig. 16: Queue Validation Dialog

Note: The names of queues cannot be translated, even using a custom language file. *Salutations* and *Signatures* have to be assigned to queues, which depend on a specific language, so translating the name makes no sense.

Sub-queue of It is possible to add the new queue under an existing one as sub-queue. This will be displayed as *Parent Queue::Child Queue*.

Group * It is possible to limit access to the selected group. The group creates a permission link between the queue and an agent or a customer user.

Unlock timeout minutes Any ticket on open, which is locked, in this queue will automatically unlock after the set amount of minutes. The value *0* (default) means tickets in this queue remain locked.

Escalation - first response time (minutes) The maximum amount of working time allowed before the agent contacts with the customer.

Note: First response time will not trigger for an email ticket or telephone ticket created by an agent.

Escalation - update time (minutes) The maximum amount of working time allowed between the agent contacts with the customer.

Escalation - solution time (minutes) The maximum amount of working time allowed until the ticket is marked as solved.

Note: Solution time will not reset if the ticket is reopened.

Follow up Option * Specify the handling of a follow up on closed tickets. Possible values:

new ticket The follow up will create a new ticket.

possible The follow up will reopen the already closed ticket.

reject The follow up will be rejected.

See also:

See *Auto Responses* chapter for more information.

Ticket lock after a follow up * Only applicable if the *Follow up Option* is set to *possible*. Locks the previously closed ticket, upon reopening, to the last owner. This ensures that a follow up for a ticket is processed by the agent that has previously handled that ticket.

> **Warning:** This does not take out-of-office into account. Use this setting with care to ensure or in combination with *Unlock timeout minutes*.

System address * Select one of the *Email Addresses* as the sender identity for this queue.

> **Note:** This is an ID in the database. Making changes to the *Email Addresses* can have adverse effects here.

Default sign key This is only active if *PGP Keys* or *S/MIME Certificates* is enabled in the *System Configuration*. Choose the key to sign emails per default.

Salutation * Select one of the defined *Salutations*.

Signature * Select one of the defined *Signatures*.

Calendar Select the calendar which defines working hours for this queue. Calendars are defined in the *System Configuration*.

Validity * Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

Chat Channel Chat channel that will be used for communication related to the tickets in this queue.

2.5 Queues Auto Responses

Communicating a change in, for example, service times, service levels or other information which would be good for a customer to know when opening a ticket can be a tedious and error-ridden task. Attempting to make sure all departments have the correct information and transmitting this to their customers poses a challenge.

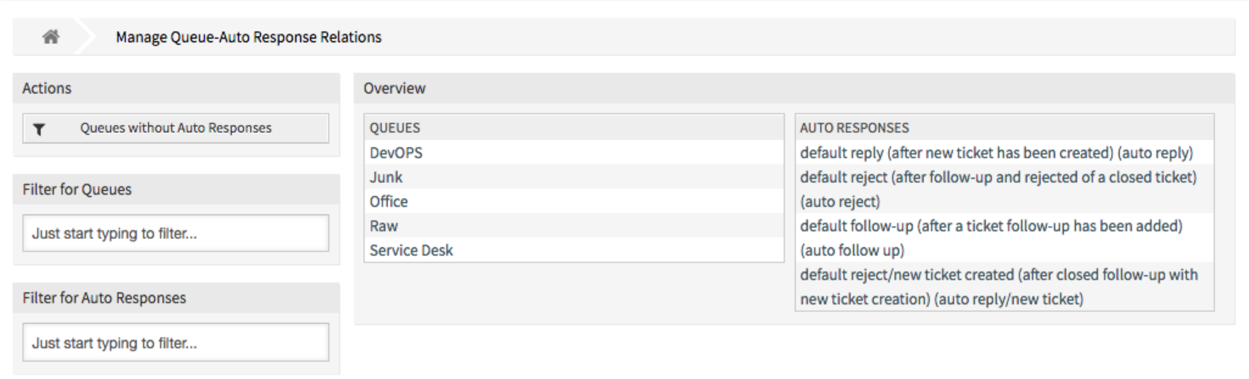
OTRS gives you the power to quickly assign the appropriate automatic responses to any queue, containing pertinent service information, ensuring this information reaches your customers before expectations aren’t reached.

Use this screen to add one or more automatic responses to one or more queues. The management screen is available in the *Queues Auto Responses* module of the *Ticket Settings* group.

2.5.1 Manage Queues Auto Responses Relations

To assign an automatic response to a queue:

1. Click on a queue in the *Queues* column.



Manage Queue-Auto Response Relations

Actions

Queues without Auto Responses

Filter for Queues

Just start typing to filter...

Filter for Auto Responses

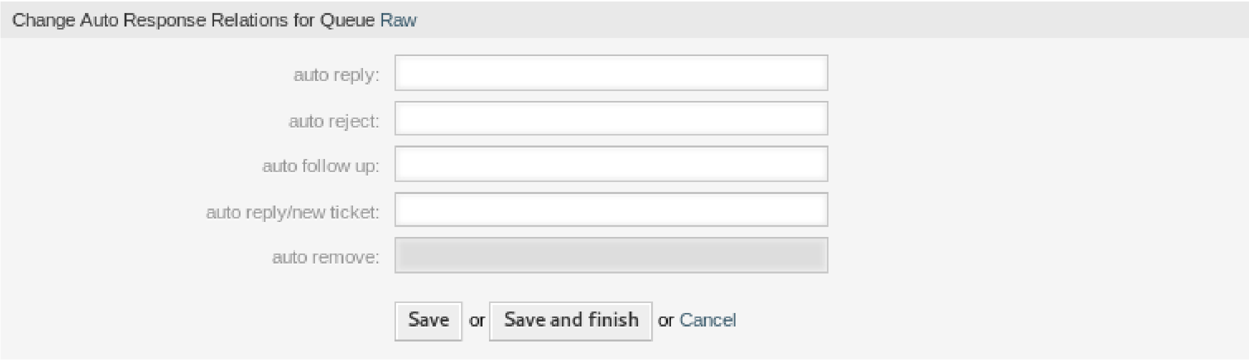
Just start typing to filter...

Overview

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

Fig. 17: Manage Queue-Auto Response Relations

2. Select the automatic responses you would like to add the queue to.
3. Click on the *Save* or *Save and finish* button.



Change Auto Response Relations for Queue Raw

auto reply:

auto reject:

auto follow up:

auto reply/new ticket:

auto remove:

Fig. 18: Change Auto Response Relations for Queue

Note: It is not possible to assign multiple queues to an automatic response by clicking on the automatic response name. A click on the automatic response will open to the *Auto Responses* screen.

Note: If several automatic responses or queues are added to the system, use the filter box to find a particular automatic response or queue by just typing the name to filter.

2.5.2 Queues Auto Responses Settings

The following settings are available when assigning some automatic responses to a queue.

auto reply This automatic response will be sent to users, when they send a message to the queue, and the message is not a follow-up message of a ticket.

auto reject This automatic response will be sent to users, when they send a message to the queue and *reject* is set as *Follow up Option* in the queue settings.

auto follow up This automatic response will be sent to users, when they send a message to the queue and *possible* is set as *Follow up Option* in the queue settings.

auto reply/new ticket This automatic response will be sent to users, when they send a message to the queue and *new ticket* is set as *Follow up Option* in the queue settings.

auto remove This automatic response will be sent to users, when the remove option is enabled.

Note: *Auto reply*, *auto reject* and *auto reply/new ticket* mutually cancel each other based on the *Queues* settings. Only one will take effect per queue.

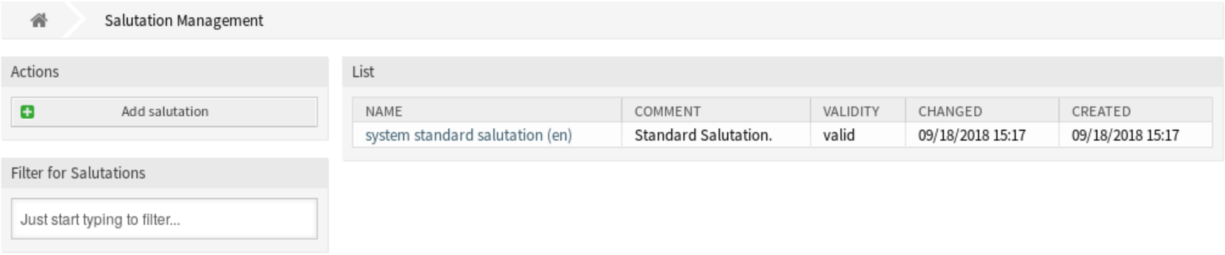
2.6 Salutations

Addressing customers must be done in a standardized way. Your customers may not always be external customers requiring a less formal greeting.

OTRS provides you with the tools needed to create a standardized communication form for any one of your queues. As defined in the *Queue Settings*: *Salutations*, *Templates*, and *Signatures* are combined to ensure a well formed standardized email communication.

Salutations can be linked to one or more *Queues*. A salutation is used only in email answers to tickets.

Use this screen to add salutations to the system. A fresh OTRS installation already contains a standard salutation. The salutation management screen is available in the *Salutations* module of the *Ticket Settings* group.



NAME	COMMENT	VALIDITY	CHANGED	CREATED
system standard salutation (en)	Standard Salutation.	valid	09/18/2018 15:17	09/18/2018 15:17

Fig. 19: Salutation Management Screen

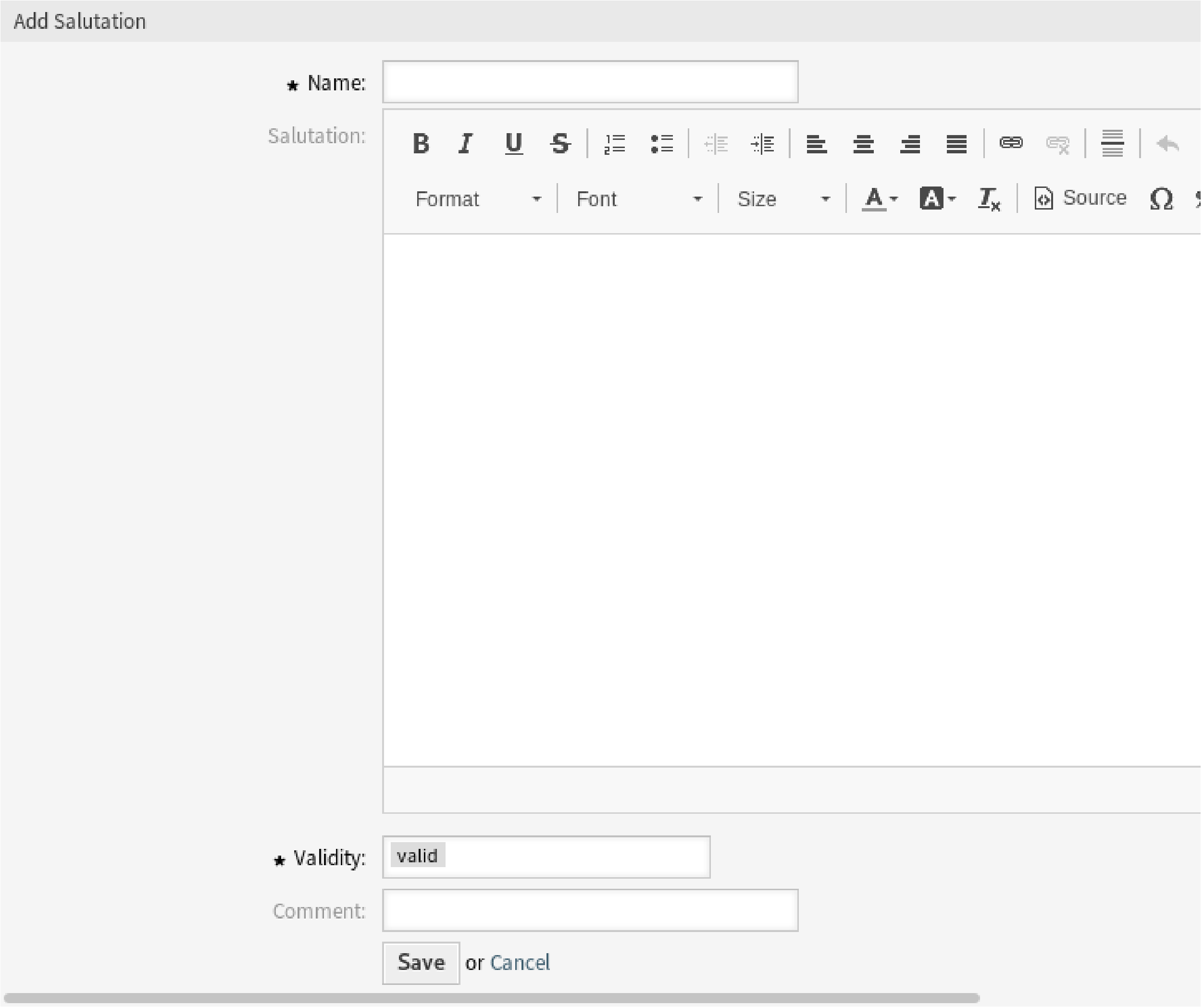
2.6.1 Manage Salutations

To add a salutation:

1. Click on the *Add Salutation* button in the left sidebar.
2. Fill in the required fields.
3. Click on the *Save* button.

Warning: Salutations can not be deleted from the system. They can only be deactivated by setting the *Validity* option to *invalid* or *invalid-temporarily*.

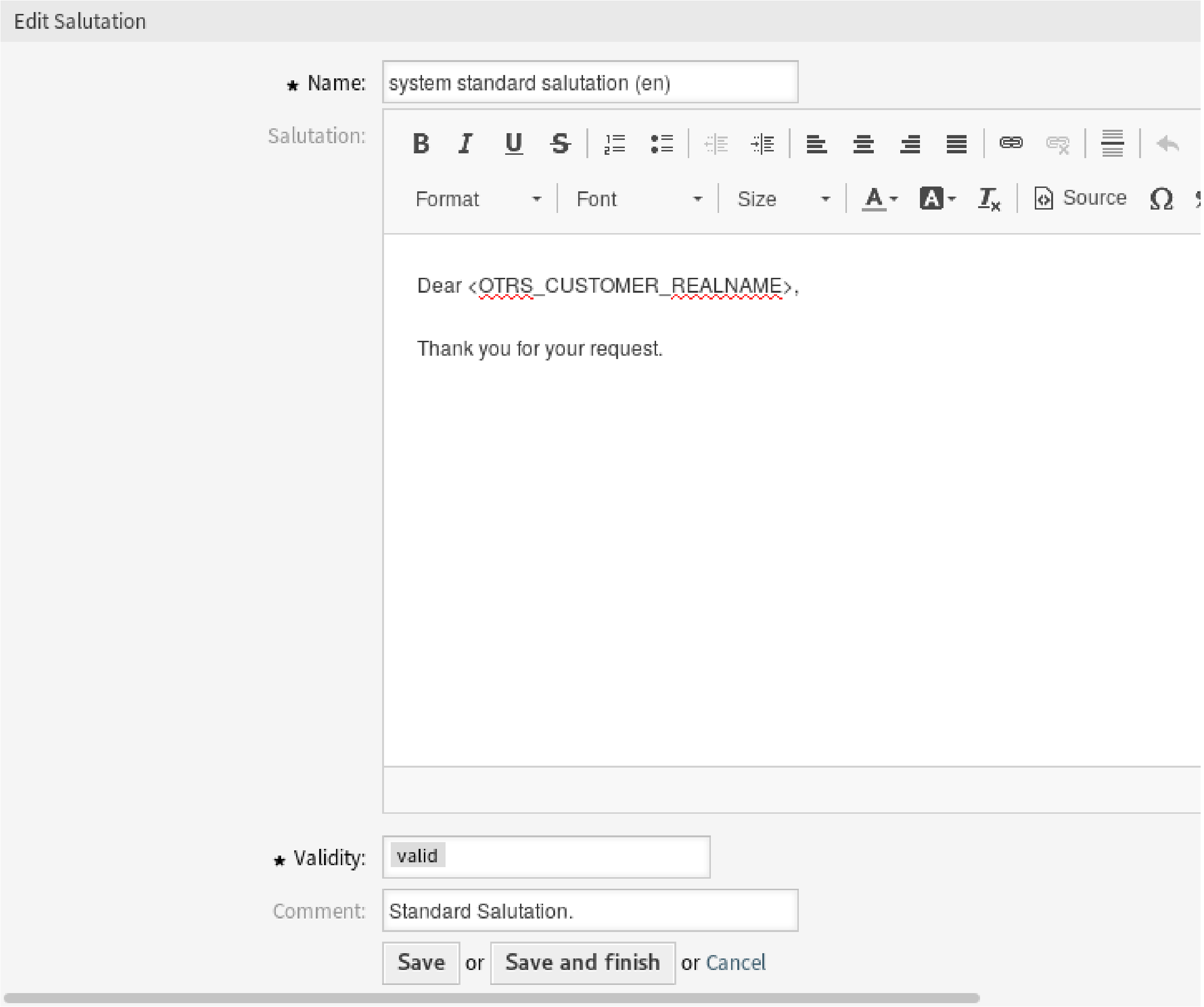
To edit a salutation:



★ Validity: valid

Fig. 20: Add Salutation Screen

1. Click on a salutation in the list of salutations.
2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.



Edit Salutation

★ Name: system standard salutation (en)

Salutation:

Dear <OTRS_CUSTOMER_REALNAME>,

Thank you for your request.

★ Validity: valid

Comment: Standard Salutation.

Fig. 21: Edit Salutation Screen

Note: If several salutations are added to the system, use the filter box to find a particular salutation by just typing the name to filter.

Warning: Before invalidating this object, please go to the *Queues* module of the *Ticket Settings* group and make sure all queues using this setting are using a valid object.

2.6.2 Salutation Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

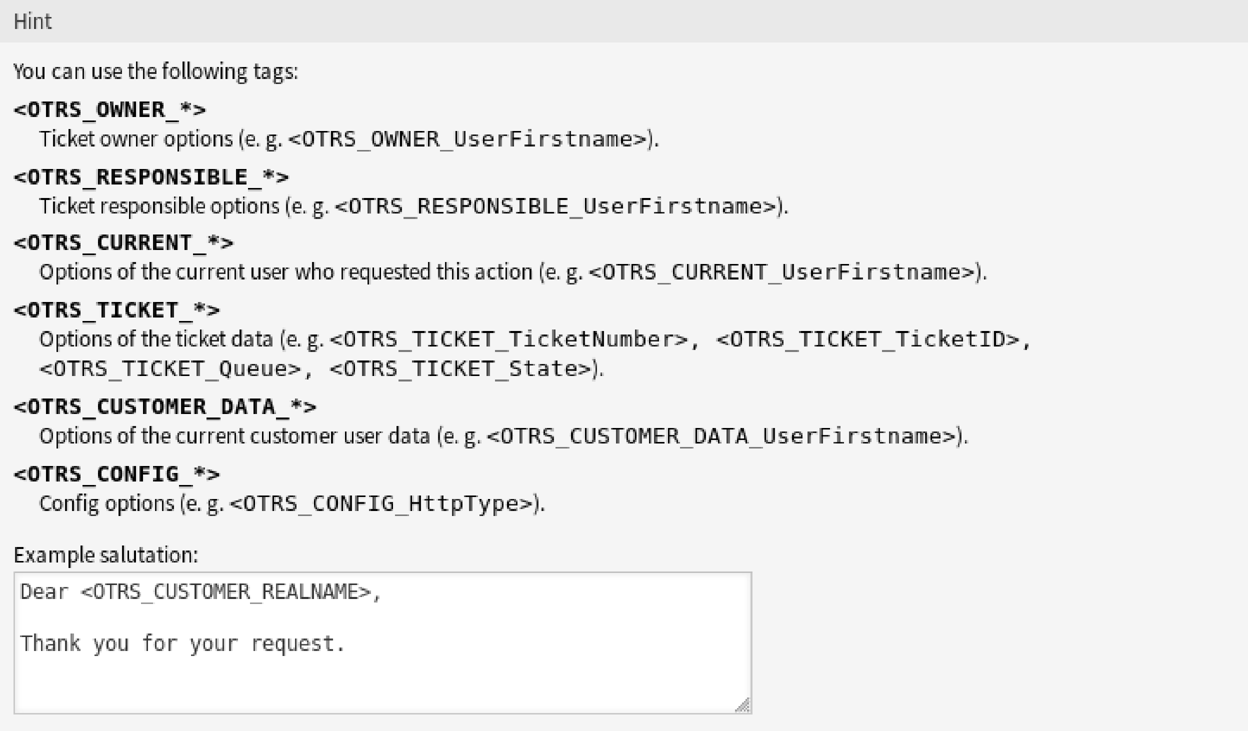
Salutation The text that will be placed to the beginning of new emails.

Validity * Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

2.6.3 Salutation Variables

Using variables in the text makes it possible to personalize messages. Variables, known as OTRS tags, are replaced by OTRS when generating the message. Find a list of available tags stems for this resource at the bottom of both add and edit screens.



Hint

You can use the following tags:

<OTRS_OWNER_*>
Ticket owner options (e. g. <OTRS_OWNER_UserFirstname>).

<OTRS_RESPONSIBLE_*>
Ticket responsible options (e. g. <OTRS_RESPONSIBLE_UserFirstname>).

<OTRS_CURRENT_*>
Options of the current user who requested this action (e. g. <OTRS_CURRENT_UserFirstname>).

<OTRS_TICKET_*>
Options of the ticket data (e. g. <OTRS_TICKET_TicketNumber>, <OTRS_TICKET_TicketID>, <OTRS_TICKET_Queue>, <OTRS_TICKET_State>).

<OTRS_CUSTOMER_DATA_*>
Options of the current customer user data (e. g. <OTRS_CUSTOMER_DATA_UserFirstname>).

<OTRS_CONFIG_*>
Config options (e. g. <OTRS_CONFIG_HttpType>).

Example salutation:

```
Dear <OTRS_CUSTOMER_REALNAME>,

Thank you for your request.
```

Fig. 22: Salutation Variables

For example, the variable `<OTRS_CUSTOMER_DATA_UserLastname>` expands to the customer’s last name to be included in something like the following.

```
Dear <OTRS_CUSTOMER_DATA_UserFirstname> <OTRS_CUSTOMER_DATA_UserLastname>,
```

This tag expands, for example to:

```
Dear Lisa Wagner,
```

The values of the following variables are translated based on the chosen language of the customer user.

```
OTRSTicket_Type
OTRSTicket_State
OTRSTicket_StateType
OTRSTicket_Lock
OTRSTicket_Priority
```

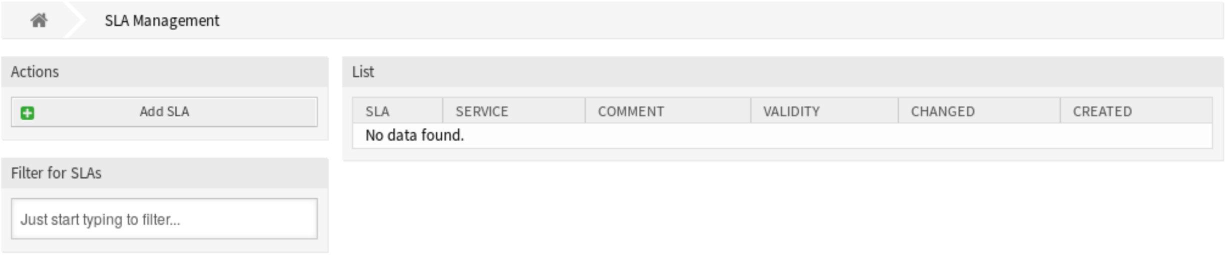
If the language is not supported the default language is applied.

2.7 Service Level Agreements

Your organization must meet the time demands of your customers. Timely service matters. Response to questions, updates on issues, and solutions must be provided in an agreed amount of time. The agent must receive notification of possible breaches, to prevent ticket escalation.

OTRS scales well with your needs and offers management of service level agreements (SLAs). OTRS provides you with the possibility to create numerous service level agreements covering all of your service and customer need. Each *SLA* can cover multiple services and define the availability of service and escalation periods.

Use this screen to add service level agreements to the system. A fresh OTRS installation doesn't contain any service level agreements by default. The service level agreement management screen is available in the *Service Level Agreements* module of the *Ticket Settings* group.



SLA	SERVICE	COMMENT	VALIDITY	CHANGED	CREATED
No data found.					

Fig. 23: Service Level Agreement Management Screen

See also:

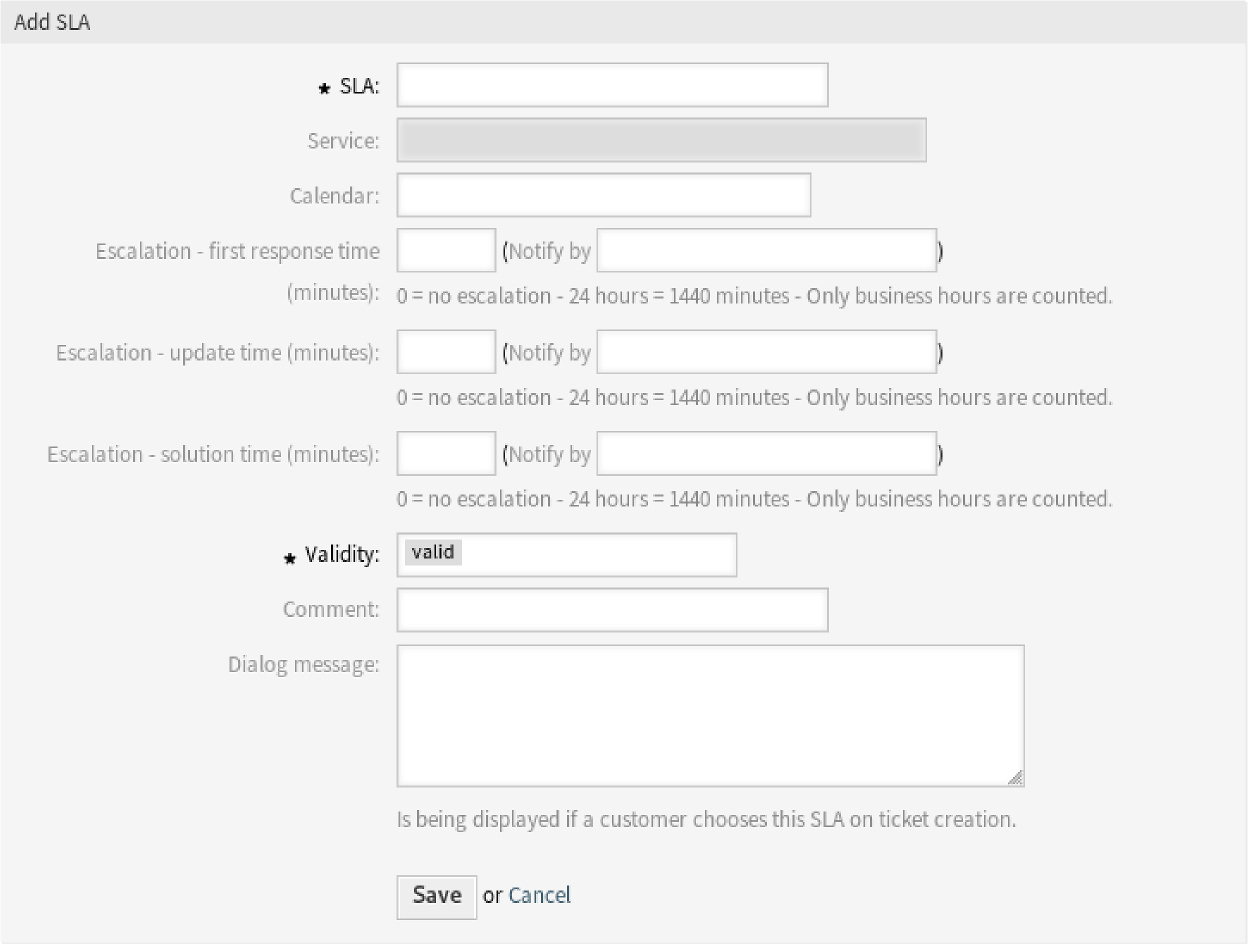
To use this feature, Ticket::Service must be activated in the *System Configuration* under the *Administration* group to be selectable in the ticket screens. You may click on the link in the warning message of the notification bar to directly jump to the configuration setting.

2.7.1 Manage Service Level Agreements

Note: Adding service level agreements requires, that at least one service is added to the system. Create services in the *Services* screen.

To add a service level agreement:

1. Click on the *Add SLA* button in the left sidebar.
2. Fill in the required fields.
3. Click on the *Save* button.



Add SLA

⋆ SLA:

Service:

Calendar:

Escalation - first response time (minutes): (Notify by)
0 = no escalation - 24 hours = 1440 minutes - Only business hours are counted.

Escalation - update time (minutes): (Notify by)
0 = no escalation - 24 hours = 1440 minutes - Only business hours are counted.

Escalation - solution time (minutes): (Notify by)
0 = no escalation - 24 hours = 1440 minutes - Only business hours are counted.

⋆ Validity: valid

Comment:

Dialog message:
Is being displayed if a customer chooses this SLA on ticket creation.

Save or Cancel

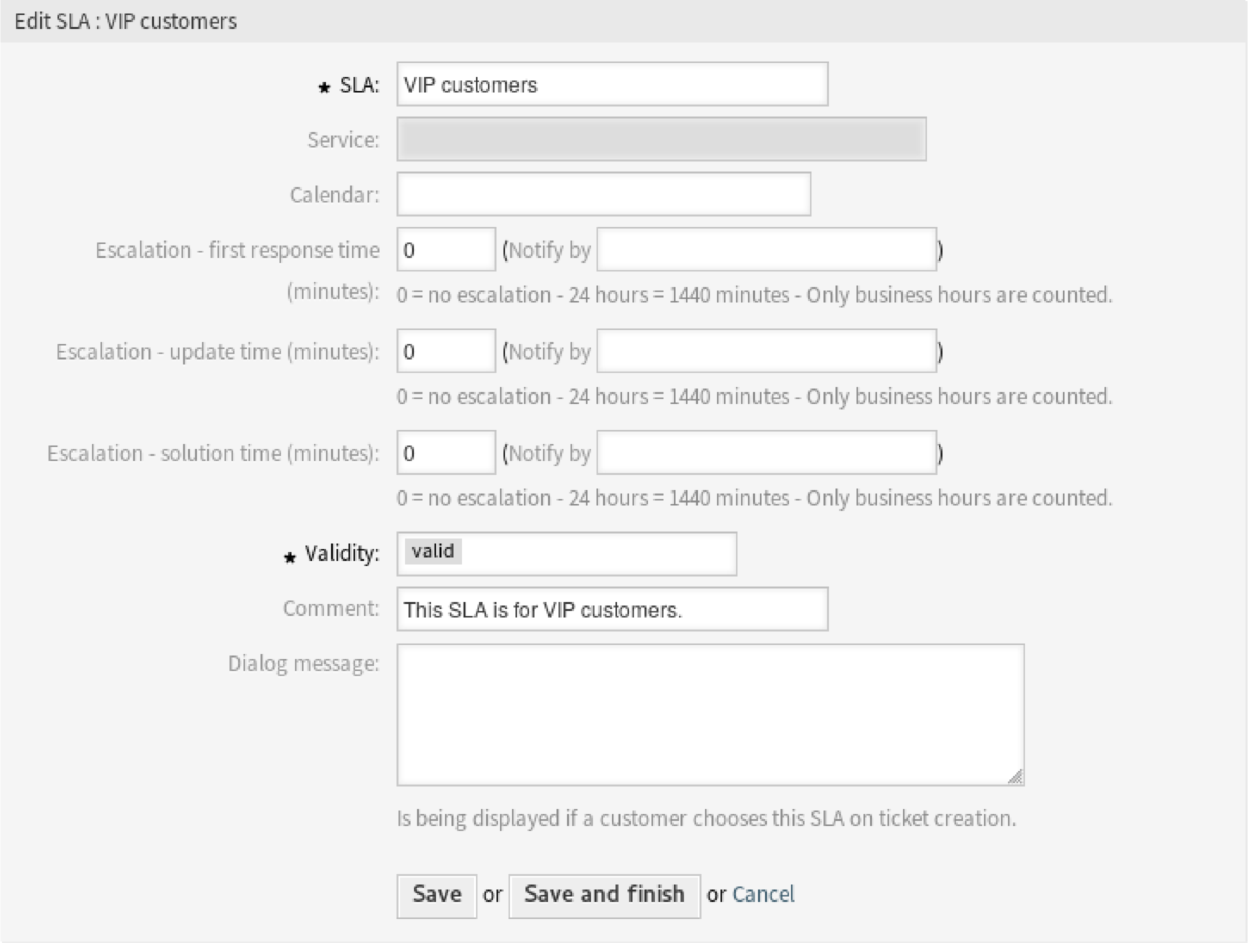
Fig. 24: Add Service Level Agreement Screen

Warning: Service level agreements can not be deleted from the system. They can only be deactivated by setting the *Validity* option to *invalid* or *invalid-temporarily*.

To edit a service level agreement:

1. Click on a service level agreement in the list of service level agreements.
2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.

Note: If several service level agreements are added to the system, use the filter box to find a particular service level agreement by just typing the name to filter.



Edit SLA : VIP customers

★ SLA: VIP customers

Service:

Calendar:

Escalation - first response time (minutes): 0 (Notify by)

0 = no escalation - 24 hours = 1440 minutes - Only business hours are counted.

Escalation - update time (minutes): 0 (Notify by)

0 = no escalation - 24 hours = 1440 minutes - Only business hours are counted.

Escalation - solution time (minutes): 0 (Notify by)

0 = no escalation - 24 hours = 1440 minutes - Only business hours are counted.

★ Validity: valid

Comment: This SLA is for VIP customers.

Dialog message:

Is being displayed if a customer chooses this SLA on ticket creation.

Fig. 25: Edit Service Level Agreement Screen

2.7.2 Service Level Agreement Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

SLA * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

> **Note:** The names of service level agreements cannot be translated, even using a custom language file. Service level agreements can use calendars, that depend on time zones. Because of this, different service level agreements have to be created for each time zone and translating their names makes no sense.

Service Select one or more of the *Services* to be assigned to this SLA.

Calendar Select the calendar which defines working hours for this queue. Calendars are defined in the *System Configuration*.

Escalation - first response time (minutes) The maximum amount of working time allowed before the agent contacts with the customer.

> **Note:** First response time will not trigger for an email ticket or telephone ticket created by an agent.

Escalation - update time (minutes) The maximum amount of working time allowed between the agent contacts with the customer.

Escalation - solution time (minutes) The maximum amount of working time allowed until the ticket is marked as solved.

> **Note:** Solution time will not reset if the ticket is reopened.

Validity * Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

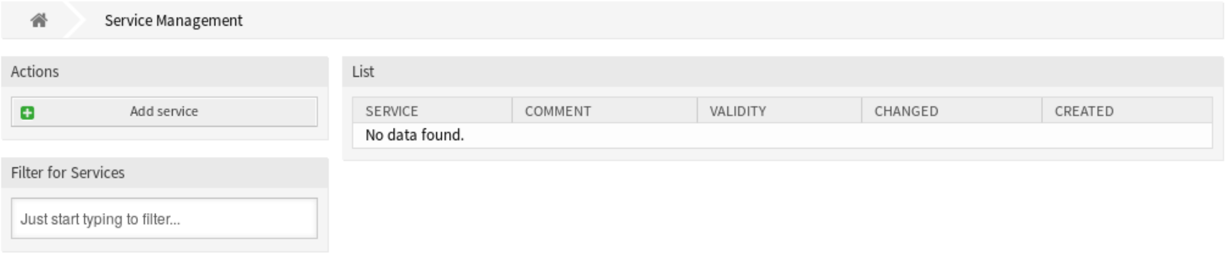
Dialog message Is being displayed if a customer chooses this SLA on ticket creation (only in the external interface).

2.8 Services

Requests to your service desk should be categorized based on affected *service* to correctly reach a team of experts to deal with the issue. Because not all your agents can deal with specific problems. Lack of experience or access to resources to fix an issue requires defining the service affected in a ticket helps to categorize the issue and target the correct teams.

OTRS allows adding all services offered to your customers. These services may be later bound to *Service Level Agreements* to ensure a timely solution based on customer-specific agreements.

Use this screen to add services to the system. A fresh OTRS installation doesn’t contain any services by default. The service management screen is available in the *Services* module of the *Ticket Settings* group.



SERVICE	COMMENT	VALIDITY	CHANGED	CREATED
No data found.				

Fig. 26: Service Management Screen

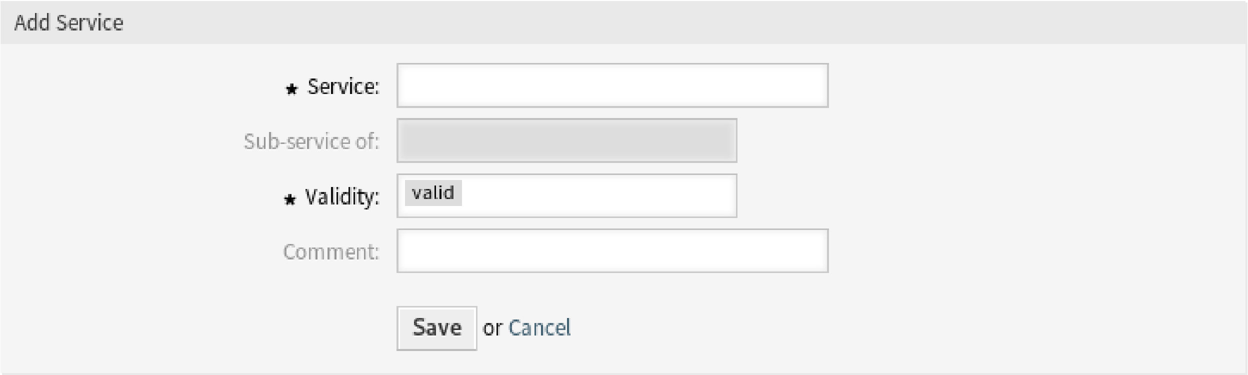
See also:

To use this feature, Ticket::Service must be activated in the *System Configuration* under the *Administration* group to be selectable in the ticket screens. You may click on the link in the warning message of the notification bar to directly jump to the configuration setting.

2.8.1 Manage Services

To add a service:

1. Click on the *Add Service* button in the left sidebar.
2. Fill in the required fields.
3. Click on the *Save* button.



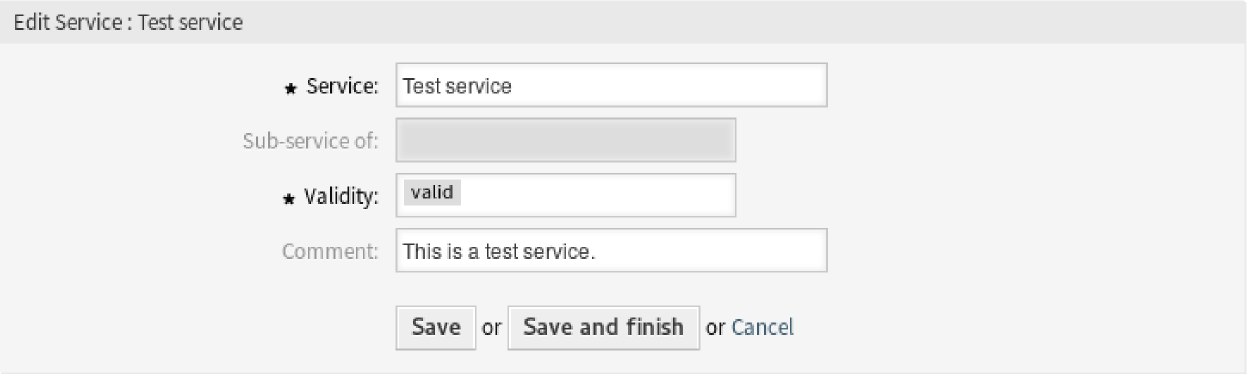
★ Validity: valid

Fig. 27: Add Service Screen

Warning: Services can not be deleted from the system. They can only be deactivated by setting the *Validity* option to *invalid* or *invalid-temporarily*.

To edit a service:

1. Click on a service in the list of services.
2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.



Edit Service : Test service

★ Service:	Test service
Sub-service of:	
★ Validity:	valid
Comment:	This is a test service.

Fig. 28: Edit Service Screen

Note: If several services are added to the system, use the filter box to find a particular service by just typing the name to filter.

Warning: Changing the name of this object should be done with care, the check only provides verification for certain settings and ignores things where the name can't be verified. Some examples are dashboard filters, access control lists (ACLs), and processes (sequence flow actions) to name a few. Documentation of your setup is key to surviving a name change.

2.8.2 Service Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Service * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Sub-service of It is possible to add the new service under an existing one as sub-service. This will be displayed as *Parent Service::Child Service*.

Validity * Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

Note: A service can only be selectable in the ticket screens, if the service is assigned to the customer user of the ticket or it is set as default service for all customer users in the *Customer Users Services* screen.

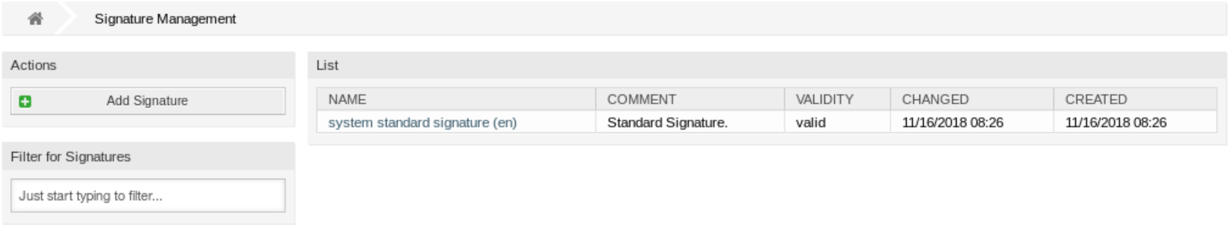
2.9 Signatures

Corporate identity and team information are essential in each communication. The name of the employee writing and other vital details like disclaimers are some examples of necessary information to include in the communication with the customer.

OTRS provides you with the same tools here, as with *Salutations*, to create a standardized communication form for any one of your queues. As defined in the *Queue Settings*: *Salutations*, *Templates*, and *Signatures* are combined to ensure a well formed standardized email communication.

Signatures can be linked to one or more *Queues*. A signature is used only in email answers to tickets.

Use this screen to add signatures to the system. A fresh OTRS installation already contains a standard signature. The signature management screen is available in the *Signatures* module of the *Ticket Settings* group.



NAME	COMMENT	VALIDITY	CHANGED	CREATED
system standard signature (en)	Standard Signature.	valid	11/16/2018 08:26	11/16/2018 08:26

Fig. 29: Signature Management Screen

2.9.1 Manage Signatures

To add a signature:

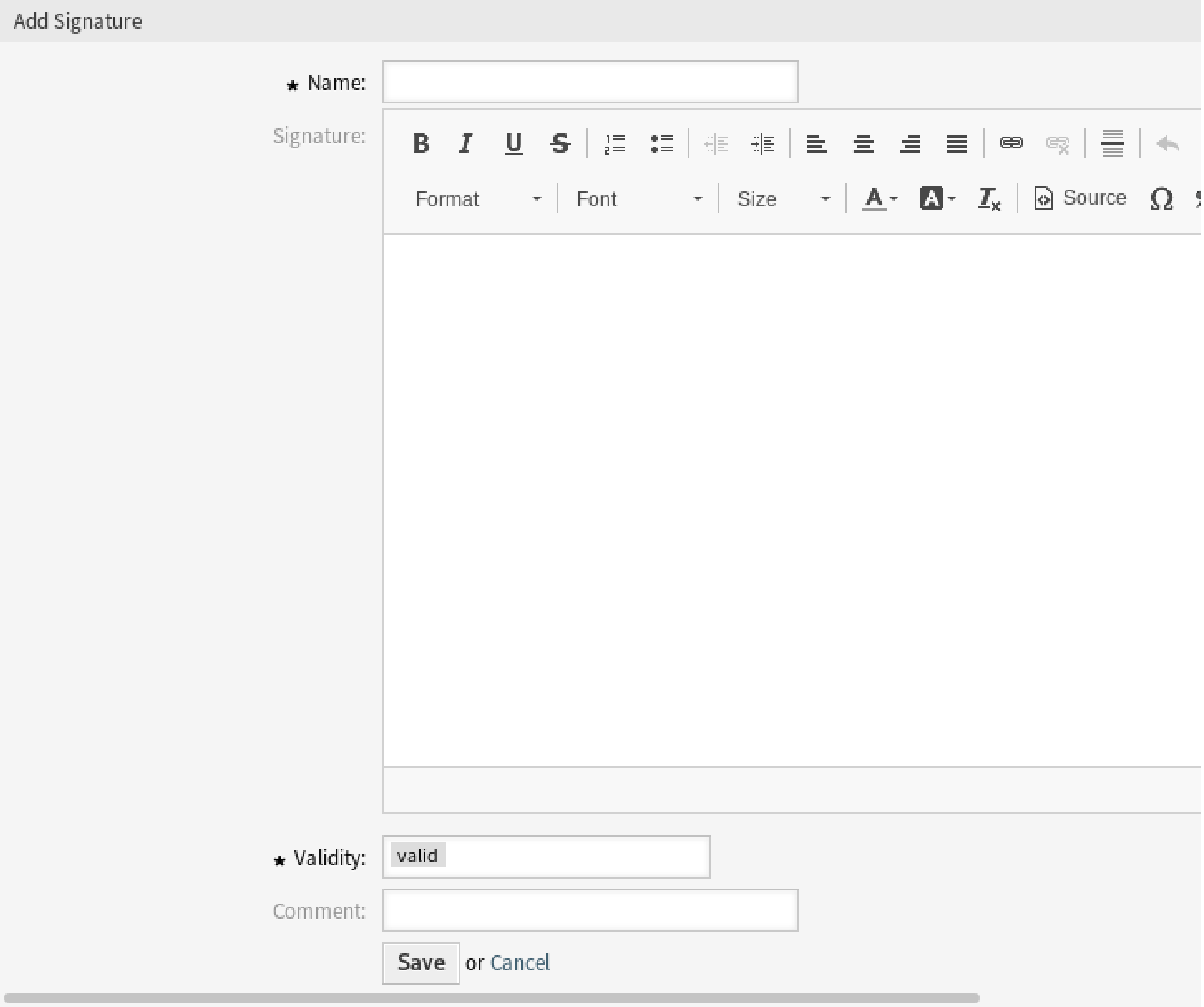
1. Click on the *Add Signature* button in the left sidebar.
2. Fill in the required fields.
3. Click on the *Save* button.

Warning: Signatures can not be deleted from the system. They can only be deactivated by setting the *Validity* option to *invalid* or *invalid-temporarily*.

To edit a signature:

1. Click on a signature in the list of signatures.
2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.

Note: If several signatures are added to the system, use the filter box to find a particular signature by just typing the name to filter.



Add Signature

⋆ Name:

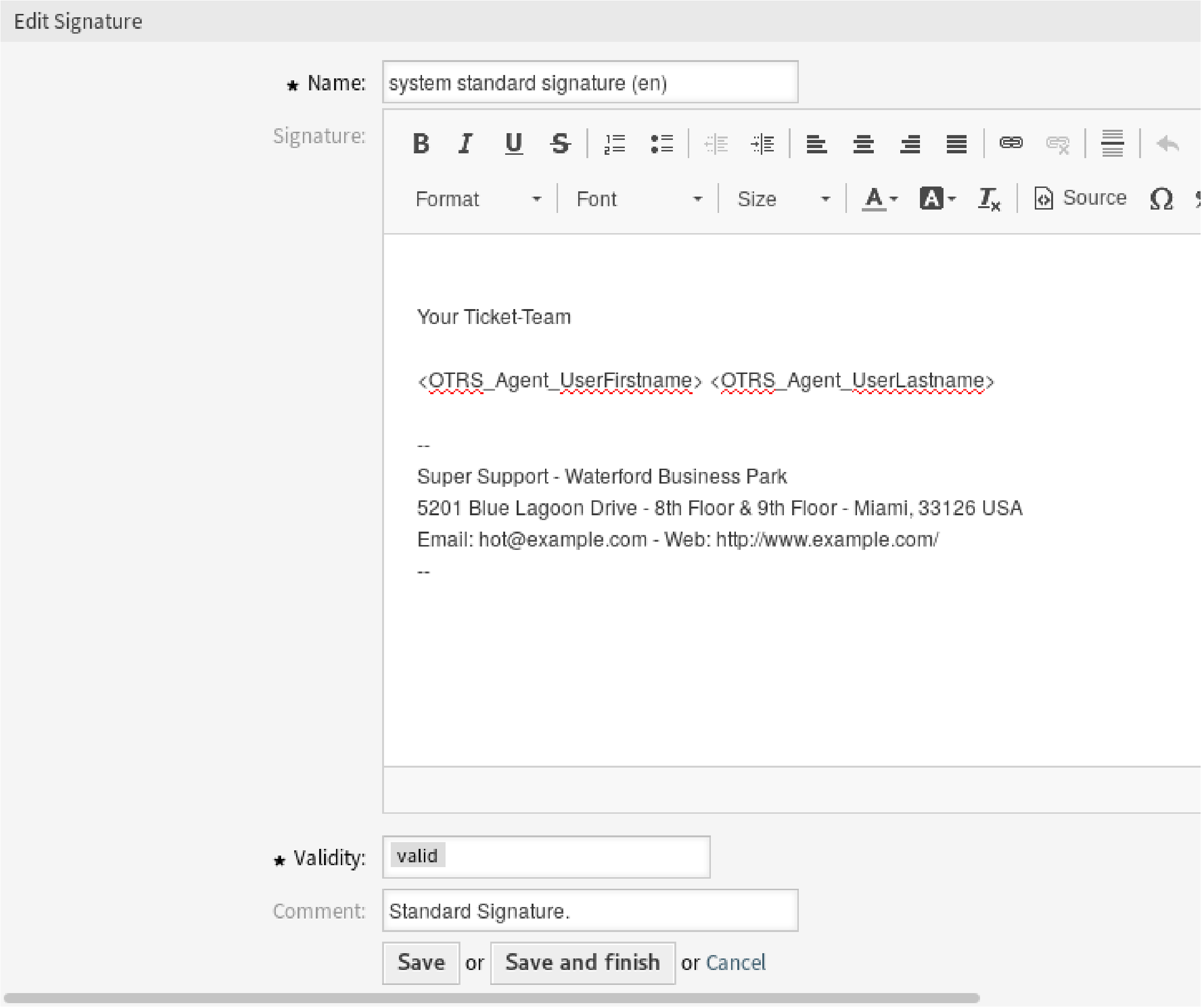
Signature:

⋆ Validity: valid

Comment:

Save or Cancel

Fig. 30: Add Signature Screen



Edit Signature

★ Name: system standard signature (en)

Signature:

Your Ticket-Team

<OTRS_Agent_UserFirstname> <OTRS_Agent_UserLastname>

--
Super Support - Waterford Business Park
5201 Blue Lagoon Drive - 8th Floor & 9th Floor - Miami, 33126 USA
Email: hot@example.com - Web: http://www.example.com/
--

★ Validity: valid

Comment: Standard Signature.

Fig. 31: Edit Signature Screen

> **Warning:** Before invalidating this object, please go to the *Queues* module of the *Ticket Settings* group and make sure all queues using this setting are using a valid object.

2.9.2 Signature Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

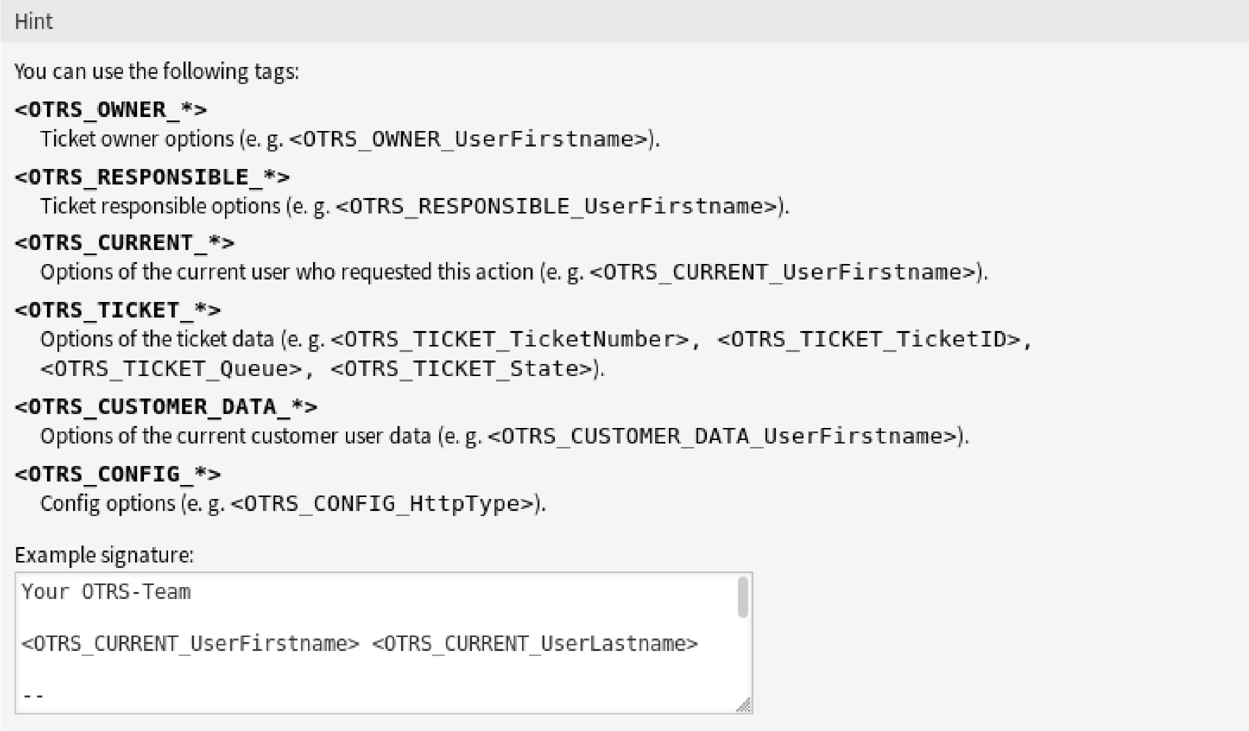
Signature The text that will be placed to the end of new emails.

Validity * Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

2.9.3 Signature Variables

Using variables in the text makes it possible to personalize messages. Variables, known as OTRS tags, are replaced by OTRS when generating the message. Find a list of available tags stems for this resource at the bottom of both add and edit screens.



Hint

You can use the following tags:

<OTRS_OWNER_*>
Ticket owner options (e. g. <OTRS_OWNER_UserFirstname>).

<OTRS_RESPONSIBLE_*>
Ticket responsible options (e. g. <OTRS_RESPONSIBLE_UserFirstname>).

<OTRS_CURRENT_*>
Options of the current user who requested this action (e. g. <OTRS_CURRENT_UserFirstname>).

<OTRS_TICKET_*>
Options of the ticket data (e. g. <OTRS_TICKET_TicketNumber>, <OTRS_TICKET_TicketID>, <OTRS_TICKET_Queue>, <OTRS_TICKET_State>).

<OTRS_CUSTOMER_DATA_*>
Options of the current customer user data (e. g. <OTRS_CUSTOMER_DATA_UserFirstname>).

<OTRS_CONFIG_*>
Config options (e. g. <OTRS_CONFIG_HttpType>).

Example signature:

```
Your OTRS-Team

<OTRS_CURRENT_UserFirstname> <OTRS_CURRENT_UserLastname>

--
```

Fig. 32: Signature Variables

For example, the variable `<OTRS_CURRENT_UserFirstname> <OTRS_CURRENT_UserLastname>` expands to the agent’ s first and last name allowing a template to include something like the following.

```
Best regards,

<OTRS_CURRENT_UserFirstname> <OTRS_CURRENT_UserLastname>
```

This tag expands, for example to:

```
Best regards,

Steven Weber
```

The values of the following variables are translated based on the chosen language of the customer user.

```
OTRSTicket_Type
OTRSTicket_State
OTRSTicket_StateType
OTRSTicket_Lock
OTRSTicket_Priority
```

If the language is not supported the default language is applied.

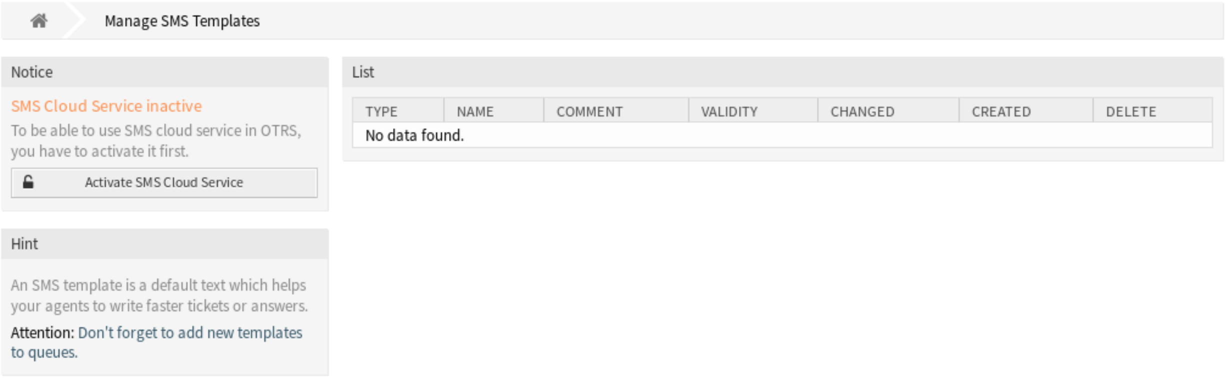
2.10 SMS Templates

An on-call duty should be alarmed about incidents on an email servers, therefore cannot get an email from OTRS. Additionally, in the case where customers have no internet access, it’ s imperative to ensure good contact.

OTRS provides SMS as a cloud service and allows, as with email, management of this communication via templates.

An SMS template is a default text which helps your agents to write faster tickets or answers.

Use this screen to add SMS templates for use in communications. A fresh OTRS installation doesn’ t contain any SMS templates by default. The SMS template management screen is available in the *SMS Templates* module of the *Ticket Settings* group.



To be able to use SMS cloud service in OTRS, you have to activate it first.

An SMS template is a default text which helps your agents to write faster tickets or answers.

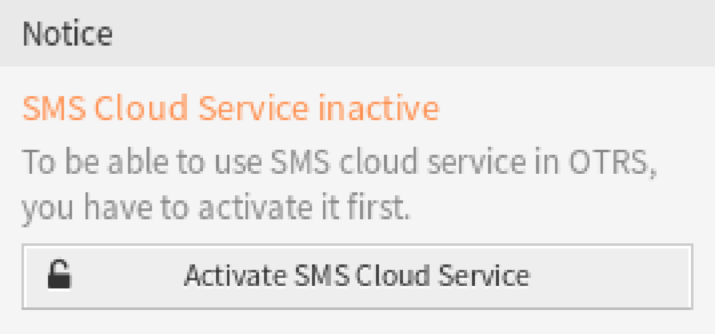
Attention: Don't forget to add new templates to queues.

TYPE	NAME	COMMENT	VALIDITY	CHANGED	CREATED	DELETE
No data found.						

Fig. 33: SMS Template Management Screen

2.10.1 Manage SMS Templates

Note: To be able to use SMS cloud service in OTRS, you have to activate it first in *Cloud Services* module.



Notice

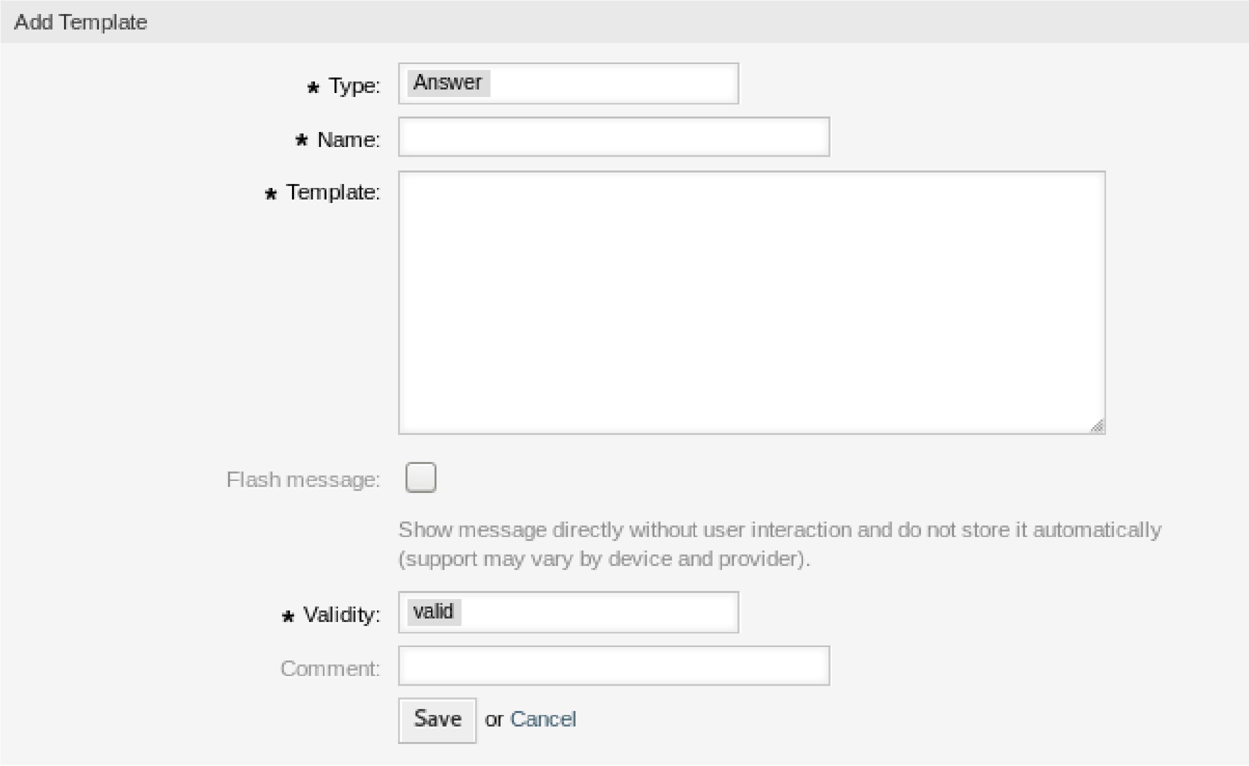
SMS Cloud Service inactive

To be able to use SMS cloud service in OTRS, you have to activate it first.

Fig. 34: Activate SMS Cloud Service

To add an SMS template:

1. Click on the *Add SMS Template* button in the left sidebar.
2. Fill in the required fields.
3. Click on the *Save* button.



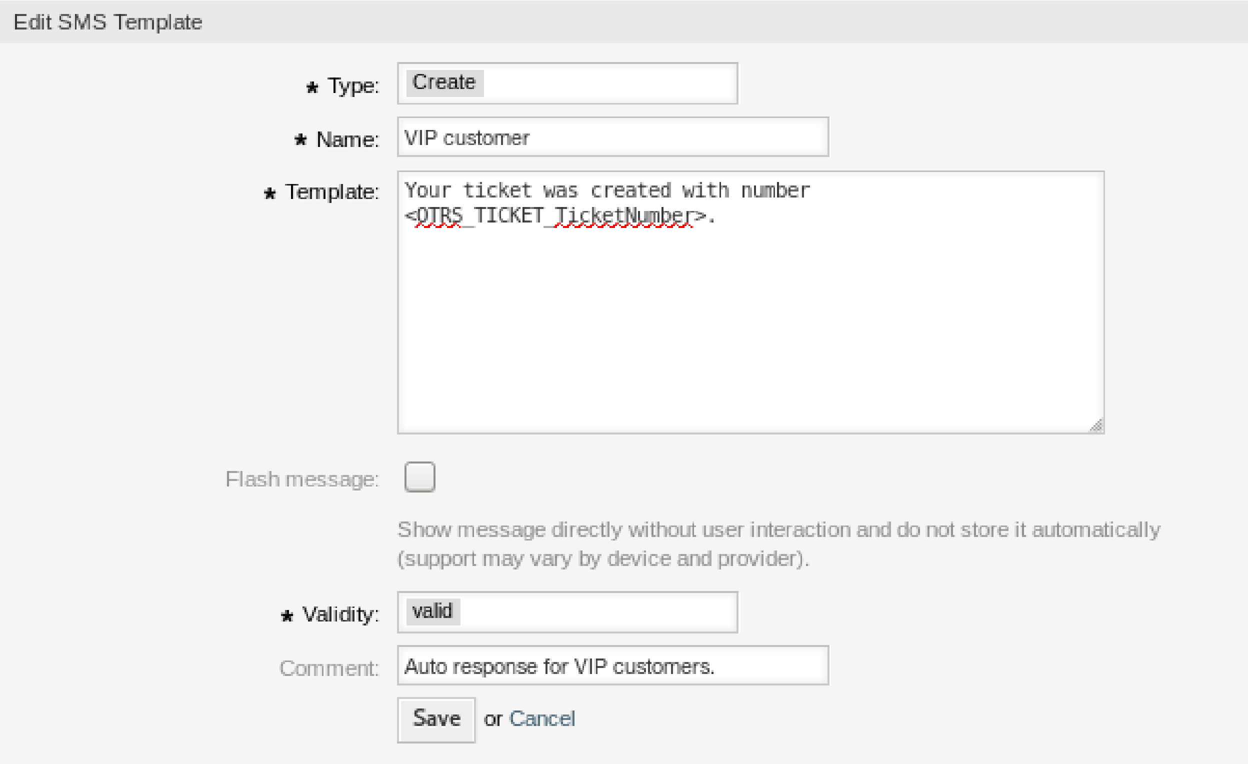
Add Template

* Type:	Answer
* Name:	
* Template:	
Flash message:	□
	Show message directly without user interaction and do not store it automatically (support may vary by device and provider).
* Validity:	valid
Comment:	

Fig. 35: Add SMS Template Screen

To edit an SMS template:

1. Click on an SMS template in the list of SMS templates.
2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.



Edit SMS Template

* Type: Create

* Name: VIP customer

* Template:

```
Your ticket was created with number
<OTRS_TICKET_TicketNumber>.
```

Flash message: □

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

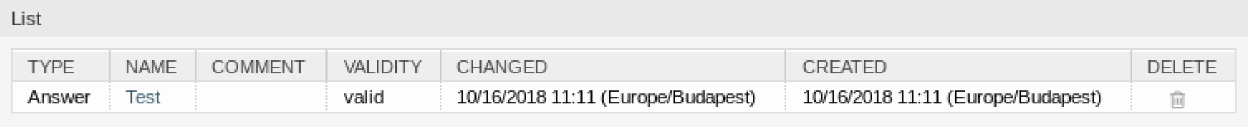
* Validity: valid

Comment: Auto response for VIP customers.

Fig. 36: Edit SMS Template Screen

To delete an SMS template:

1. Click on the trash icon in the list of SMS templates.
2. Click on the *Confirm* button.



List

TYPE	NAME	COMMENT	VALIDITY	CHANGED	CREATED	DELETE
Answer	Test		valid	10/16/2018 11:11 (Europe/Budapest)	10/16/2018 11:11 (Europe/Budapest)	🗑

Fig. 37: Delete SMS Template Screen

Note: If several SMS templates are added to the system, a filter box is useful to find a particular SMS template by just typing to filter.

2.10.2 SMS Template Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Type * There are different kind of SMS templates that are used for different purposes. A template can be:

> **Answer** To be used as a ticket response with *Reply via SMS* article action of the ticket detail view.
>
> **Create** To be used for a new SMS ticket.
>
> **SMSOutbound** To be used for sending a new SMS to a customer user from within the ticket detail view.

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Template The body of the email sent to the users.

> **Warning:** The maximum length of text message that you can send is 918 characters. However, if you send more than 160 characters then your message will be broken down in to chunks of 153 characters before being sent to the recipient’s handset. We recommend keeping the messages to less than 160 characters.

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

Validity * Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

2.10.3 SMS Template Variables

Using variables in the text makes it possible to personalize messages. Variables, known as OTRS tags, are replaced by OTRS when generating the message. Find a list of available tags stems for this resource at the bottom of both add and edit screens.

For example, the variable `<OTRS_TICKET_TicketNumber>` expands to the ticket number allowing an SMS template to include something like the following.

```
Ticket#<OTRS_TICKET_TicketNumber> has been raised in <OTRS_Ticket_Queue>.
```

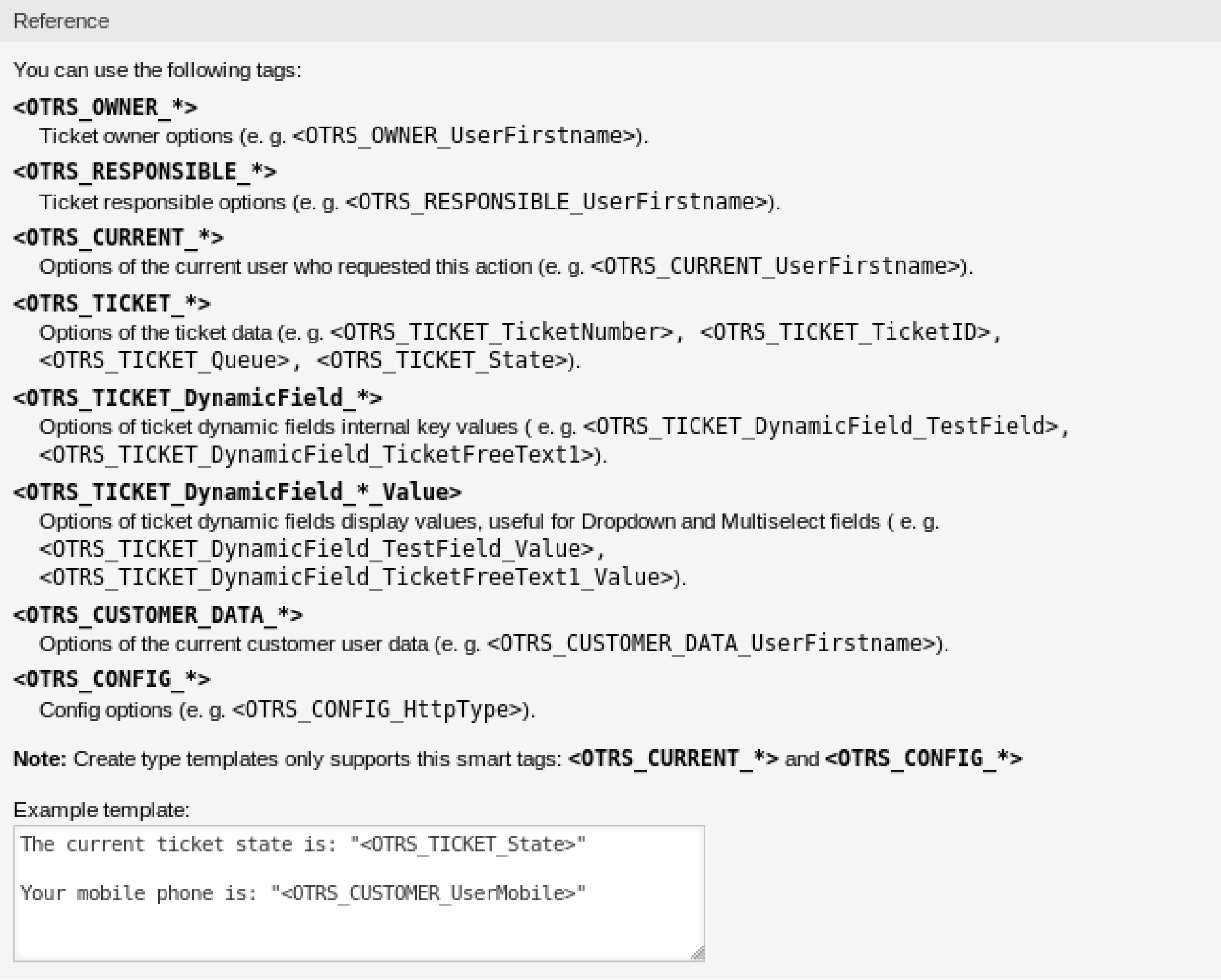
This tag expands, for example to:

```
Ticket#2018101042000012 has been raised in Postmaster.
```

The values of the following variables are translated based on the chosen language of the customer user.

```
OTRSTicket_Type
OTRSTicket_State
OTRSTicket_StateType
OTRSTicket_Lock
OTRSTicket_Priority
```

If the language is not supported the default language is applied.



Reference

You can use the following tags:

<OTRS_OWNER_*>
Ticket owner options (e. g. `<OTRS_OWNER_UserFirstname>`).

<OTRS_RESPONSIBLE_*>
Ticket responsible options (e. g. `<OTRS_RESPONSIBLE_UserFirstname>`).

<OTRS_CURRENT_*>
Options of the current user who requested this action (e. g. `<OTRS_CURRENT_UserFirstname>`).

<OTRS_TICKET_*>
Options of the ticket data (e. g. `<OTRS_TICKET_TicketNumber>, <OTRS_TICKET_TicketID>, <OTRS_TICKET_Queue>, <OTRS_TICKET_State>`).

<OTRS_TICKET_DynamicField_*>
Options of ticket dynamic fields internal key values (e. g. `<OTRS_TICKET_DynamicField_TestField>, <OTRS_TICKET_DynamicField_TicketFreeText1>`).

<OTRS_TICKET_DynamicField_*_Value>
Options of ticket dynamic fields display values, useful for Dropdown and Multiselect fields (e. g. `<OTRS_TICKET_DynamicField_TestField_Value>, <OTRS_TICKET_DynamicField_TicketFreeText1_Value>`).

<OTRS_CUSTOMER_DATA_*>
Options of the current customer user data (e. g. `<OTRS_CUSTOMER_DATA_UserFirstname>`).

<OTRS_CONFIG_*>
Config options (e. g. `<OTRS_CONFIG_HttpType>`).

Note: Create type templates only supports this smart tags: **<OTRS_CURRENT_*>** and **<OTRS_CONFIG_*>**

Example template:

```
The current ticket state is: "<OTRS_TICKET_State>"

Your mobile phone is: "<OTRS_CUSTOMER_UserMobile>"
```

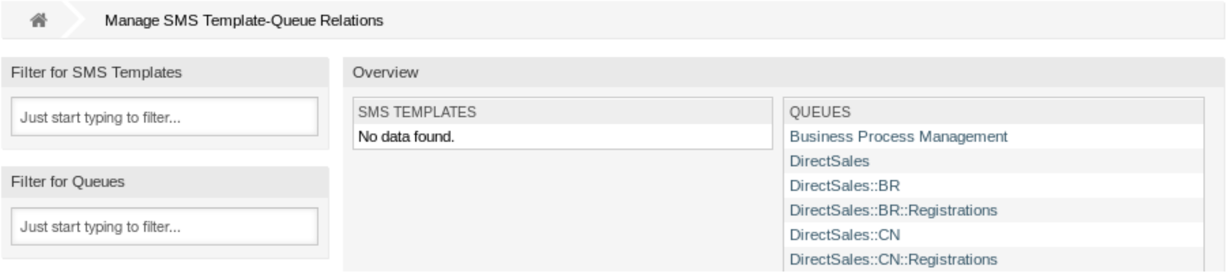
Fig. 38: SMS Template Variables

2.11 SMS Templates   Queues

Communicating a change to a template, or distribution of a new communication requirement, can be tedious and error-ridden because it requires that all teams pull the information and publish it appropriately.

OTRS gives you the power to quickly assign the appropriate SMS templates to any queue, containing pertinent ticket information, ensuring this information reaches your customers and agents.

Use this screen to add one or more SMS templates to one or more queues. The management screen is available in the *SMS Templates   Queues* module of the *Ticket Settings* group.



Manage SMS Template-Queue Relations

Filter for SMS Templates

Just start typing to filter...

Filter for Queues

Just start typing to filter...

Overview

SMS TEMPLATES
No data found.

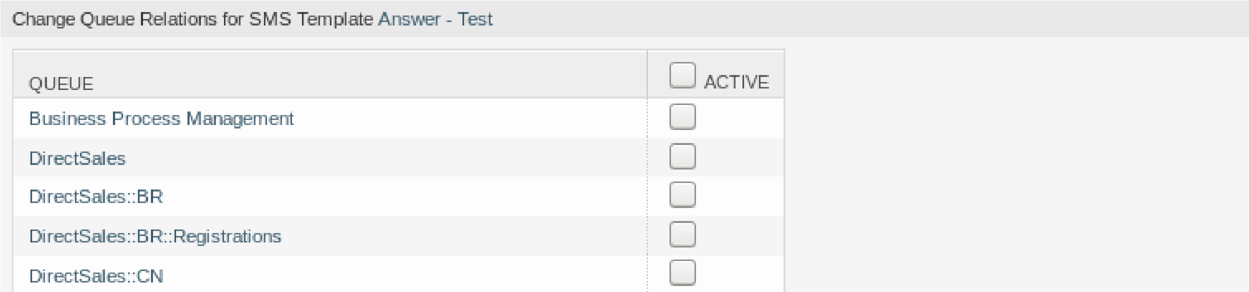
QUEUES
Business Process Management
DirectSales
DirectSales::BR
DirectSales::BR::Registrations
DirectSales::CN
DirectSales::CN::Registrations

Fig. 39: Manage SMS Template-Queue Relations

2.11.1 Manage SMS Templates   Queues Relations

To assign an SMS template to one or more queues:

1. Click on an SMS template in the *SMS Templates* column.
2. Select the queues you would like to add the SMS template to.
3. Click on the *Save* or *Save and finish* button.



Change Queue Relations for SMS Template Answer - Test

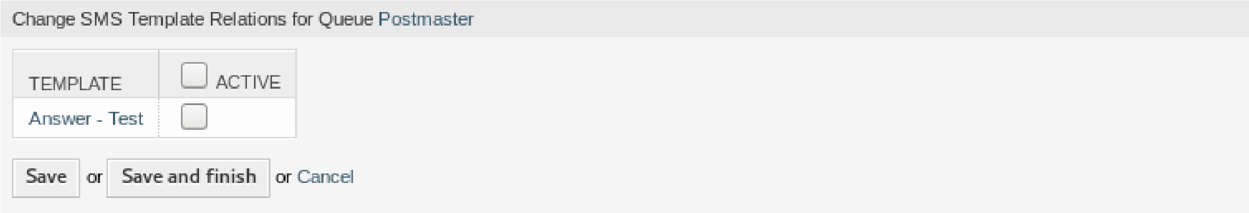
QUEUE	□ ACTIVE
Business Process Management	□
DirectSales	□
DirectSales::BR	□
DirectSales::BR::Registrations	□
DirectSales::CN	□

Fig. 40: Change Queue Relations for SMS Template

To assign a queue to one or more SMS templates:

1. Click on a queue in the *Queues* column.
2. Select the SMS templates you would like to add the queue to.
3. Click on the *Save* or *Save and finish* button.

Note: If several SMS templates or queues are added to the system, use the filter box to find a particular SMS template or queue by just typing the name to filter.



Change SMS Template Relations for Queue Postmaster

TEMPLATE	□ ACTIVE
Answer - Test	□

Fig. 41: Change SMS Template Relations for Queue

Multiple SMS templates or queues can be assigned in both screens at the same time. Additionally clicking on an SMS template or clicking on a queue in the relations screen will open the *Edit SMS Template* screen or the *Edit Queue* screen accordingly.

Warning: Accessing a queue or an SMS template provides no back link to the relations screen.

2.12 States

Active tracking of tickets leads to a better sense of workload and provides metrics as a key performance indicator. Sorting tasks and setting appointments can help to level-off the workload and keep your service desk running.

OTRS uses ticket states to ensure that your agents always know which tickets are being attended to and which not. Additionally, detailed reports on the states of your tickets can be provided by ticket search or reports and personalized sorting is possible using dashboards and queue and service overviews.

Nine states are pre-defined. More states can be added, but the default states are enough to get you going and mostly enough for any situation.

The default states are the following:

closed successful A ticket is complete. The customer received a solution which worked.

closed unsuccessful A ticket is complete. The customer received no solution or the solution was not appropriate.

merged The ticket content is found in a different ticket.

new The ticket is created by the customer without contact with an agent.

open The ticket is currently in progress. Customer and agent are in contact with each another.

pending auto close+ The ticket will be set as *closed successful* upon reaching the pending time.

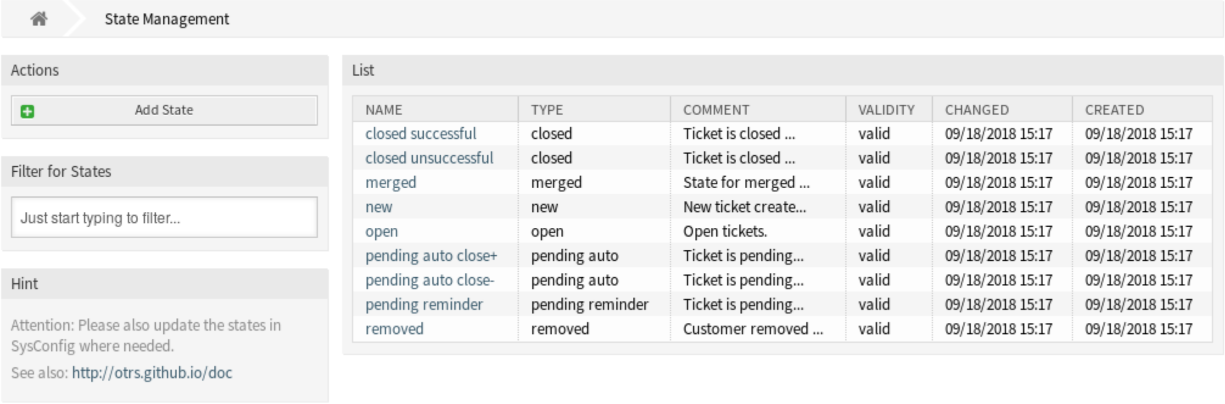
pending auto close- The ticket will be set as *closed unsuccessful* upon reaching the pending time.

pending reminder The ticket should be worked on again upon reaching the pending time.

removed The ticket has been removed from the system.

Note: Pending jobs are checked per default every two hours and forty-five minutes. This time is a static time, which means the times are 02:45, 04:45, 06:45 and so on. The job can be run more often or seldom and are configured in the *System Configuration* module of the *Administration* group.

Use this screen to add states to the system. A fresh OTRS installation contains several states by default. The state management screen is available in the *States* module of the *Ticket Settings* group.



Attention: Please also update the states in SysConfig where needed.

See also: http://otrs.github.io/doc

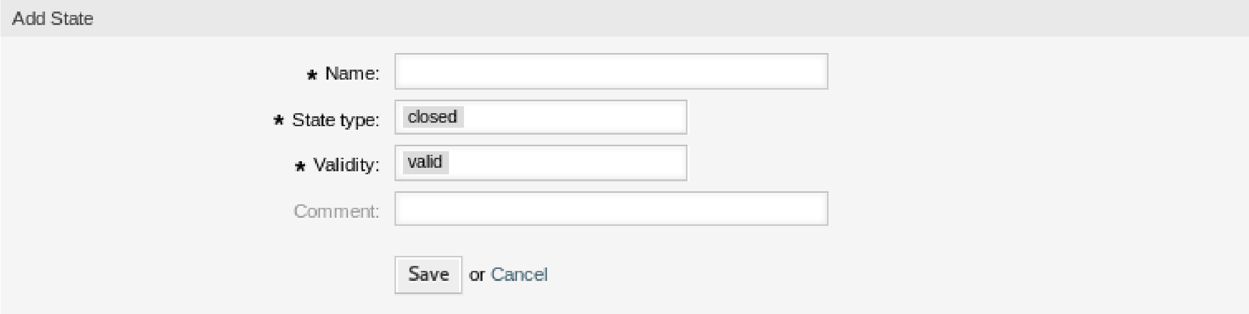
NAME	TYPE	COMMENT	VALIDITY	CHANGED	CREATED
closed successful	closed	Ticket is closed ...	valid	09/18/2018 15:17	09/18/2018 15:17
closed unsuccessful	closed	Ticket is closed ...	valid	09/18/2018 15:17	09/18/2018 15:17
merged	merged	State for merged ...	valid	09/18/2018 15:17	09/18/2018 15:17
new	new	New ticket create...	valid	09/18/2018 15:17	09/18/2018 15:17
open	open	Open tickets.	valid	09/18/2018 15:17	09/18/2018 15:17
pending auto close+	pending auto	Ticket is pending...	valid	09/18/2018 15:17	09/18/2018 15:17
pending auto close-	pending auto	Ticket is pending...	valid	09/18/2018 15:17	09/18/2018 15:17
pending reminder	pending reminder	Ticket is pending...	valid	09/18/2018 15:17	09/18/2018 15:17
removed	removed	Customer removed ...	valid	09/18/2018 15:17	09/18/2018 15:17

Fig. 42: State Management Screen

2.12.1 Manage States

To add a state:

1. Click on the *Add State* button in the left sidebar.
2. Fill in the required fields.
3. Click on the *Save* button.



∗ Name:

∗ State type: closed

∗ Validity: valid

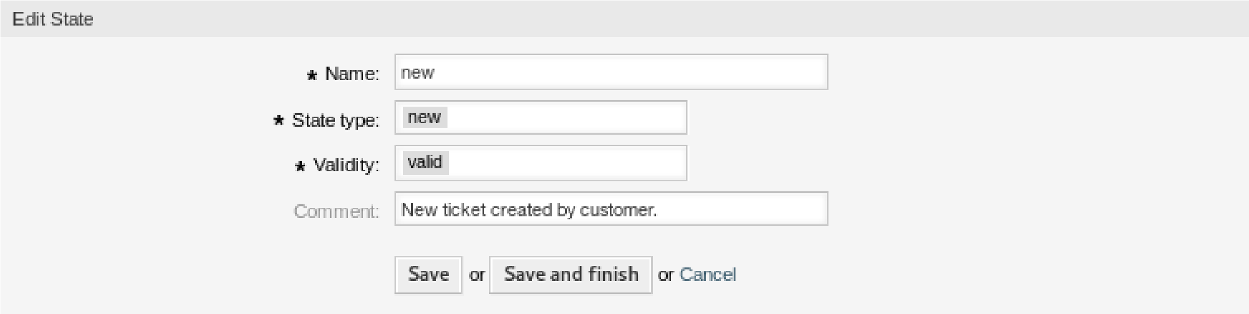
Comment:

Fig. 43: Add State Screen

> **Warning:** States can not be deleted from the system. They can only be deactivated by setting the *Validity* option to *invalid* or *invalid-temporarily*.

To edit a state:

1. Click on a state in the list of states.
2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.


Edit State

★ Name: new

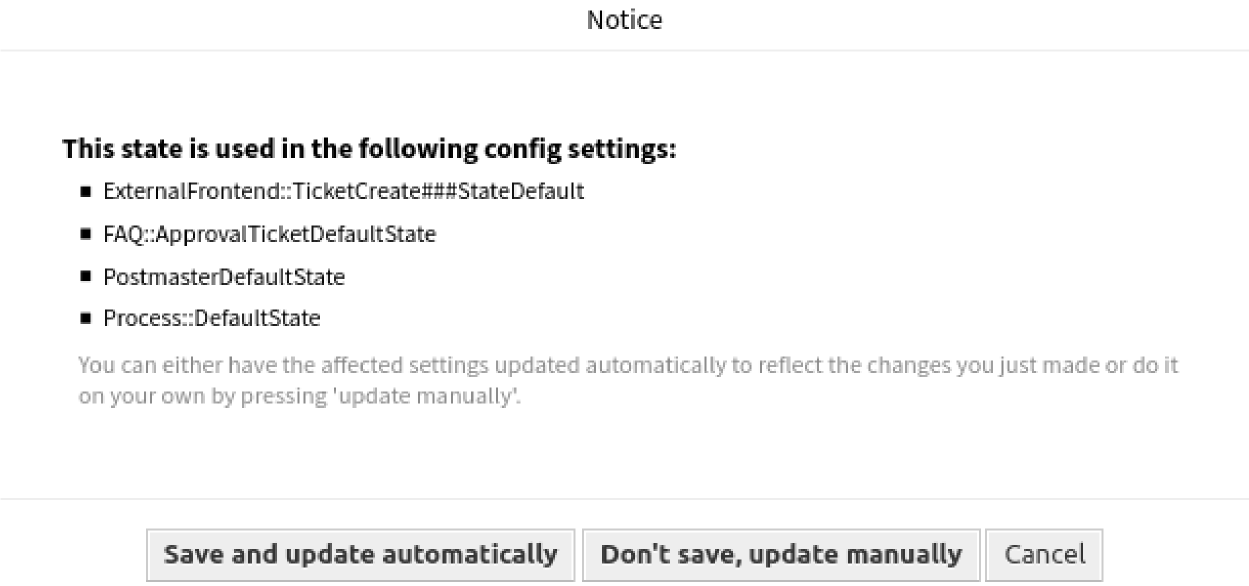
★ State type: new

★ Validity: valid

Comment: New ticket created by customer.

Fig. 44: Edit State Screen

Note: If several states are added to the system, use the filter box to find a particular state by just typing the name to filter.


Notice

This state is used in the following config settings:

- ExternalFrontend::TicketCreate###StateDefault
- FAQ::ApprovalTicketDefaultState
- PostmasterDefaultState
- Process::DefaultState

You can either have the affected settings updated automatically to reflect the changes you just made or do it on your own by pressing 'update manually'.

Fig. 45: State Validation Dialog

If you change the name of a state which is used in the system configuration, a validation check will warn you and give you the following options:

Save and update automatically Apply the change and also update the affected settings.

Don't save, update manually Apply the change, but don't update the affected settings. The updates need to be done manually.

Cancel Cancel the action.

Warning: Changing the name of this object should be done with care, the check only provides verification for certain settings and ignores things where the name can't be verified. Some examples are dashboard filters, access control lists (ACLs), and processes (sequence flow actions) to name a few.

Documentation of your setup is key to surviving a name change.

2.12.2 State Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

State type * Every state is linked to a type, which needs to be specified if a new state is created or an existing one is edited. The following types are available:

- closed
- merged
- new
- open
- pending auto
- pending reminder
- removed

Note: State types are predefined and cannot be changed in the software due to their special mechanics. When adding new states for *pending auto* and *pending reminder* state types you must make further configurations in the *System Configuration* module of the *Administration* group.

Validity * Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

See also:

The following configuration options are relevant and noteworthy when managing states:

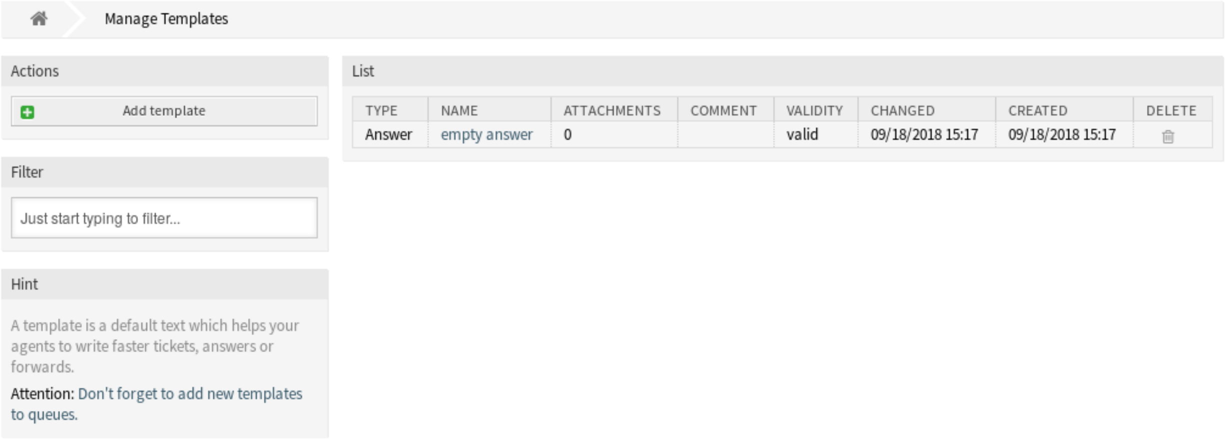
- Daemon::SchedulerCronTaskManager::Task###TicketPendingCheck
- Ticket::StateAfterPending

2.13 Templates

Providing the correct and consistent answer all the time regardless of employee or knowledge-level is important to maintain a professional appearance to your customers. Additionally, speed in sending standard answers is key to wading through the masses of requests in growing service desks.

OTRS templates offer you a variety of ways to deal with standardizing communications and help to pre-define texts so that the customer always receives the same level and quality of service from all agents.

Use this screen to add templates for use in communications. A fresh OTRS installation already contains a template by default. The template management screen is available in the *Templates* module of the *Ticket Settings* group.



TYPE	NAME	ATTACHMENTS	COMMENT	VALIDITY	CHANGED	CREATED	DELETE
Answer	empty answer	0		valid	09/18/2018 15:17	09/18/2018 15:17	🗑

A template is a default text which helps your agents to write faster tickets, answers or forwards.

Attention: Don't forget to add new templates to queues.

Fig. 46: Template Management Screen

2.13.1 Manage Templates

Note: To add attachments to a template, it needs to create the attachment first in the *Attachments* screen.

To add a template:

1. Click on the *Add Template* button in the left sidebar.
2. Fill in the required fields.
3. Click on the *Save* button.

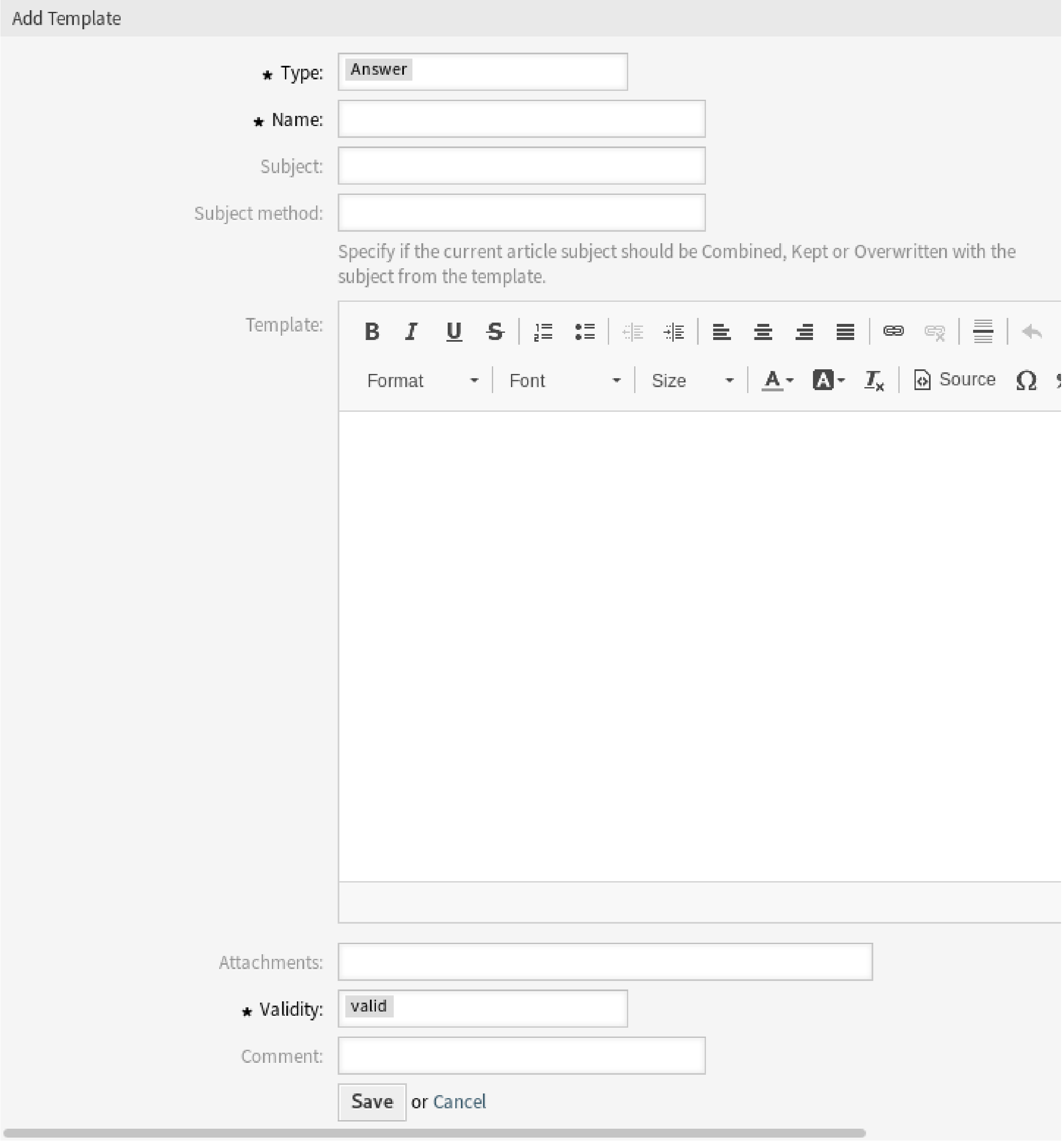
To edit a template:

1. Click on a template in the list of templates.
2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.

To delete a template:

1. Click on the trash icon in the list of templates.
2. Click on the *Confirm* button.

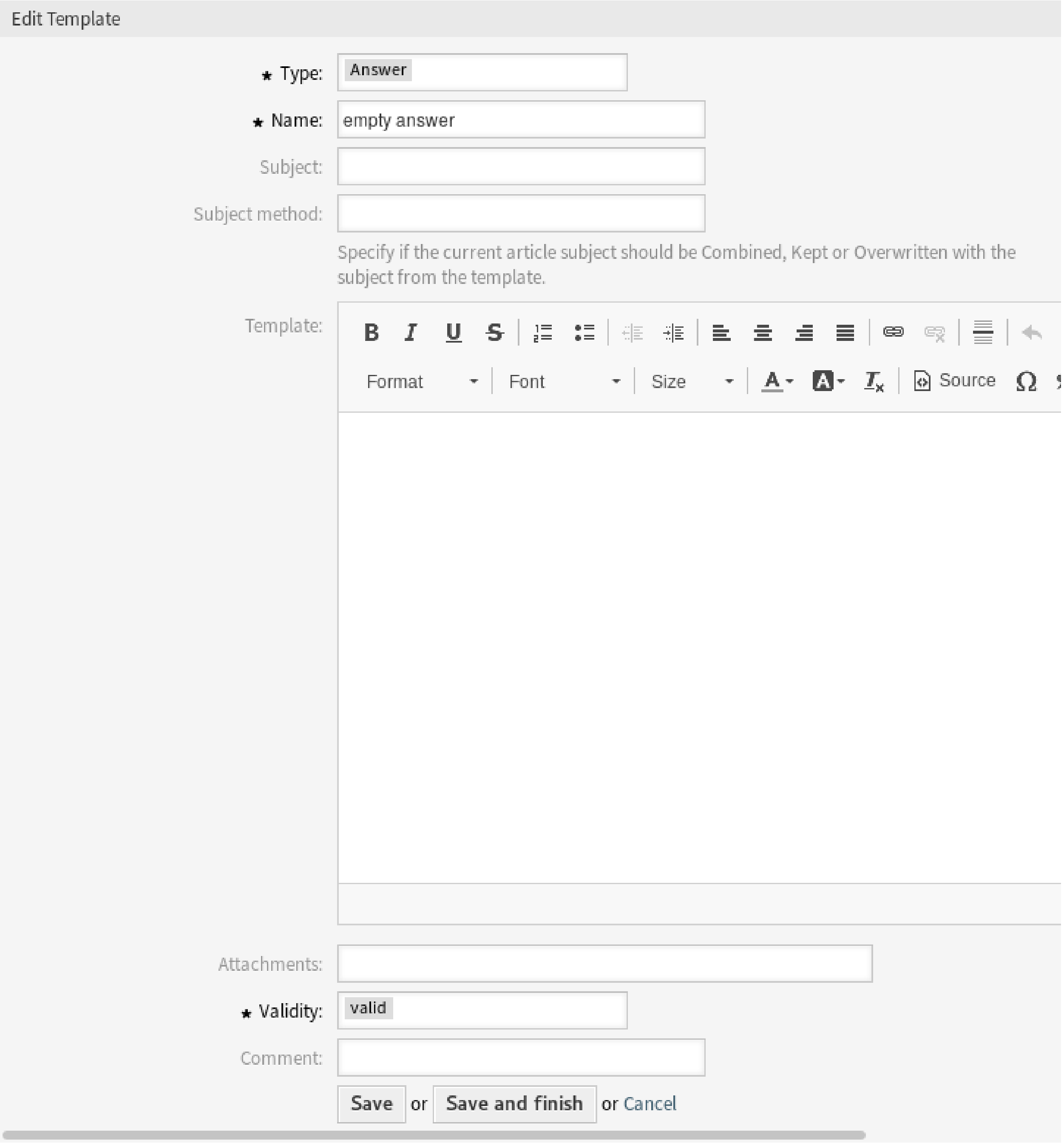
Note: If several templates are added to the system, a filter box is useful to find a particular template by just typing to filter.



Add Template

★ Type:	Answer
★ Name:	
Subject:	
Subject method:	
	Specify if the current article subject should be Combined, Kept or Overwritten with the subject from the template.
Template:	
Attachments:	
★ Validity:	valid
Comment:	

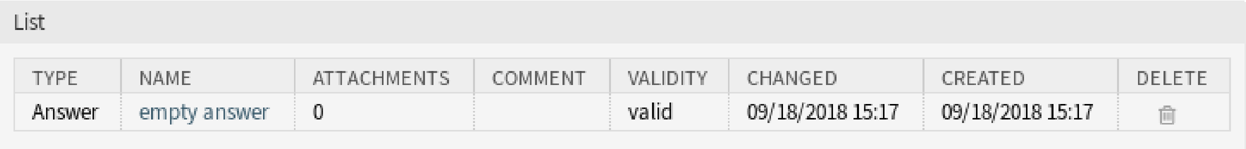
Fig. 47: Add Template Screen



Edit Template

⭑ Type:	Answer
⭑ Name:	empty answer
Subject:	
Subject method:	
	Specify if the current article subject should be Combined, Kept or Overwritten with the subject from the template.
Template:	
Attachments:	
⭑ Validity:	valid
Comment:	

Fig. 48: Edit Template Screen



List

TYPE	NAME	ATTACHMENTS	COMMENT	VALIDITY	CHANGED	CREATED	DELETE
Answer	empty answer	0		valid	09/18/2018 15:17	09/18/2018 15:17	🗑

Fig. 49: Delete Template Screen

2.13.2 Template Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Type * There are different kind of templates that are used for different purposes. A template can be:

- **Answer** To be used as a ticket response or reply.
- **Create** To be used for new phone or email ticket.
- **Email** To be used for writing an email to a customer user or to someone else.
- **Forward** To be used to forward an article to someone else.
- **Note** To be used to create notes.
- **Phone call** To be used for inbound and outbound phone calls.
- **Process dialog** To be used in process management to pre-fill text of articles in user task activity dialogs.

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Subject The subject of the article added by the template.

Subject method Specify how the current article subject should be dealt with. The following methods are available:

- **Combine** The template subject will be added after the current article subject.
- **Keep** The current article subject will be kept.
- **Overwrite** The current article subject will be replaced with the template subject.

Template The body of the article added by the template.

Attachments It is possible to add one or more attachments to this template. Attachments can be added in the *Attachments* screen.

Validity * Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

2.13.3 Template Variables

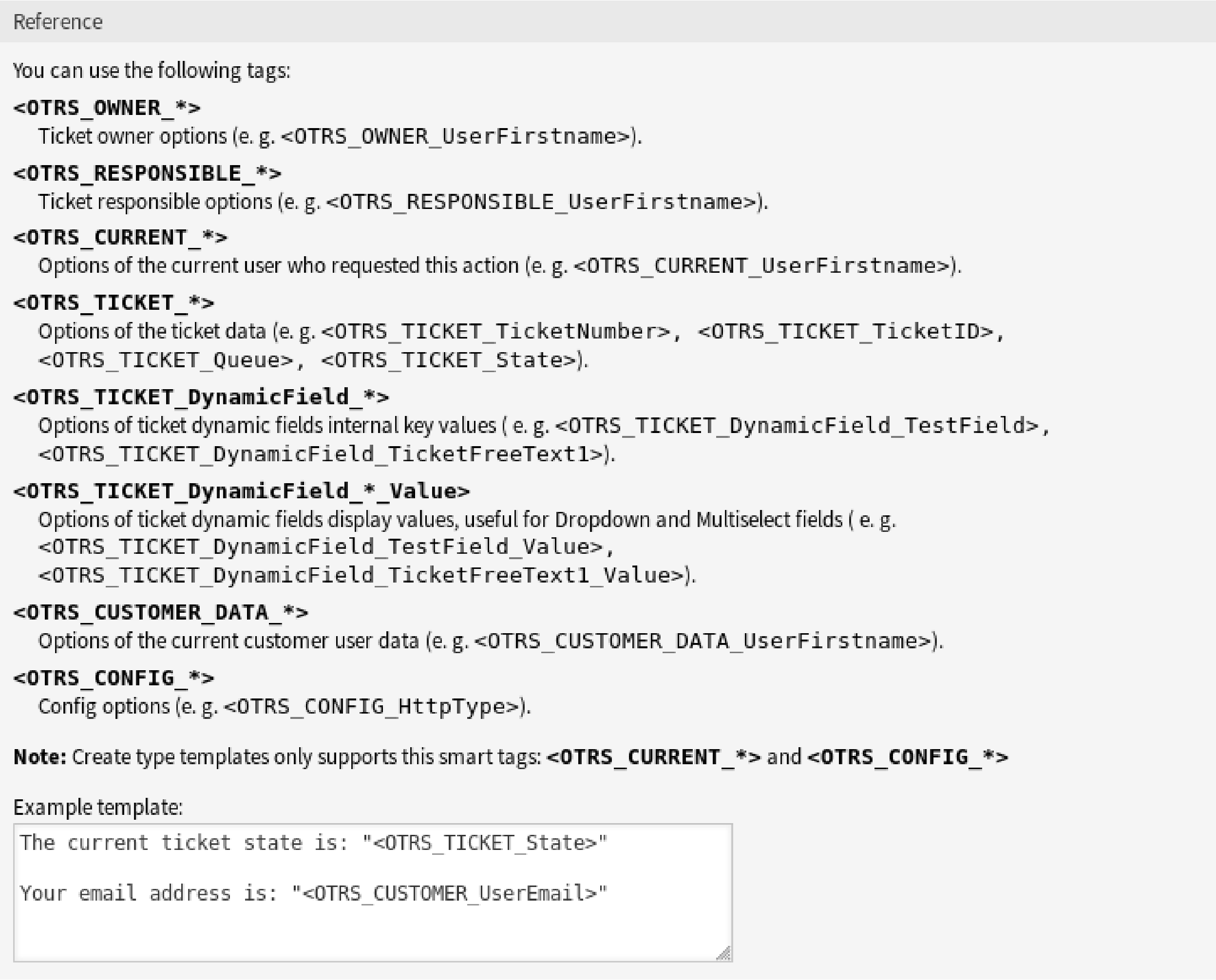
Using variables in the text makes it possible to personalize messages. Variables, known as OTRS tags, are replaced by OTRS when generating the message. Find a list of available tags stems for this resource at the bottom of both add and edit screens.

For example, the variable `<OTRS_TICKET_TicketNumber>` expands to the ticket number allowing a template to include something like the following.

```
Ticket#<OTRS_TICKET_TicketNumber>
```

This tag expands, for example to:

```
Ticket#2018101042000012
```



Reference

You can use the following tags:

<OTRS_OWNER_*>
Ticket owner options (e. g. `<OTRS_OWNER_UserFirstname>`).

<OTRS_RESPONSIBLE_*>
Ticket responsible options (e. g. `<OTRS_RESPONSIBLE_UserFirstname>`).

<OTRS_CURRENT_*>
Options of the current user who requested this action (e. g. `<OTRS_CURRENT_UserFirstname>`).

<OTRS_TICKET_*>
Options of the ticket data (e. g. `<OTRS_TICKET_TicketNumber>, <OTRS_TICKET_TicketID>, <OTRS_TICKET_Queue>, <OTRS_TICKET_State>`).

<OTRS_TICKET_DynamicField_*>
Options of ticket dynamic fields internal key values (e. g. `<OTRS_TICKET_DynamicField_TestField>, <OTRS_TICKET_DynamicField_TicketFreeText1>`).

<OTRS_TICKET_DynamicField_*_Value>
Options of ticket dynamic fields display values, useful for Dropdown and Multiselect fields (e. g. `<OTRS_TICKET_DynamicField_TestField_Value>, <OTRS_TICKET_DynamicField_TicketFreeText1_Value>`).

<OTRS_CUSTOMER_DATA_*>
Options of the current customer user data (e. g. `<OTRS_CUSTOMER_DATA_UserFirstname>`).

<OTRS_CONFIG_*>
Config options (e. g. `<OTRS_CONFIG_HttpType>`).

Note: Create type templates only supports this smart tags: **<OTRS_CURRENT_*>** and **<OTRS_CONFIG_*>**

Example template:

```
The current ticket state is: "<OTRS_TICKET_State>"

Your email address is: "<OTRS_CUSTOMER_UserEmail>"
```

Fig. 50: Template Variables

The values of the following variables are translated based on the chosen language of the customer user.

```
OTRSTicket_Type
OTRSTicket_State
OTRSTicket_StateType
OTRSTicket_Lock
OTRSTicket_Priority
```

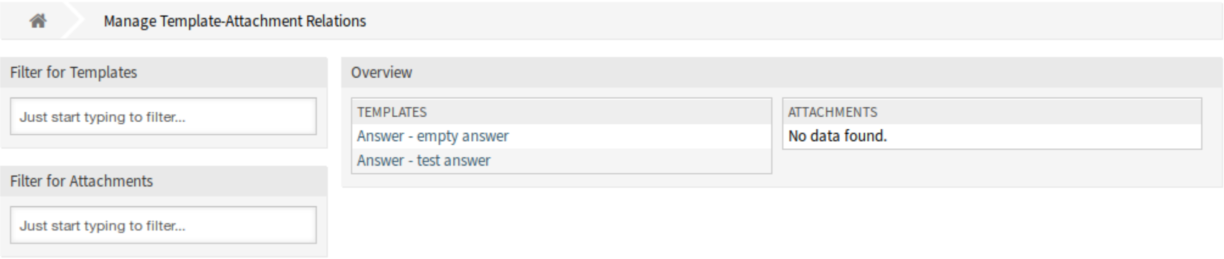
If the language is not supported the default language is applied.

2.14 Templates   Attachments

Making changes to standard attachments is sometimes overwhelming. The question is, “Where were the attachments used?” or “How to quickly update them across the board?”. It’s also important to know who are using which attachments, before updating them. A new attachment can have multiple uses within your organization.

OTRS empowers you to manage this providing you an overview to manage the 1:n relationships and quickly identify the templates using your attachments.

Use this screen to add one or more attachments to one or more templates. To use this function, at least one attachment and one template need to have been added to the system. The management screen is available in the *Templates   Attachments* module of the *Ticket Settings* group.


Manage Template-Attachment Relations
Filter for Templates
Just start typing to filter...
Filter for Attachments
Just start typing to filter...
Overview

TEMPLATES	ATTACHMENTS
Answer - empty answer	No data found.
Answer - test answer	

Fig. 51: Manage Template-Attachment Relations

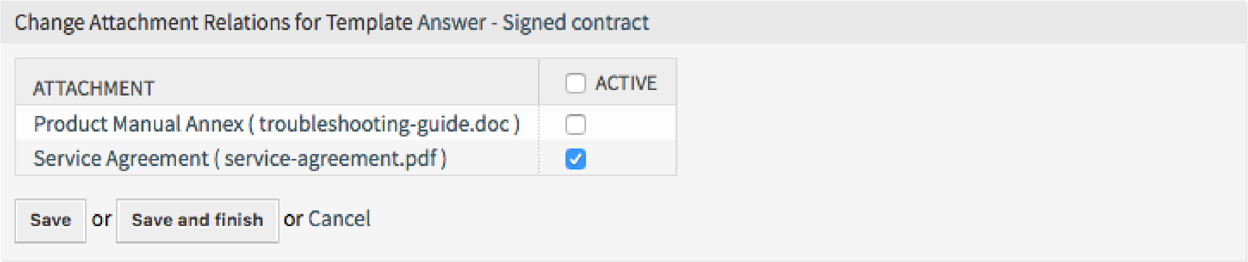
2.14.1 Manage Templates   Attachments Relations

To add some attachments to a template:

1. Click on a template in the *Templates* column.
2. Select the attachments you would like to add to the template.
3. Click on the *Save* or *Save and finish* button.

To assign an attachment to some templates:

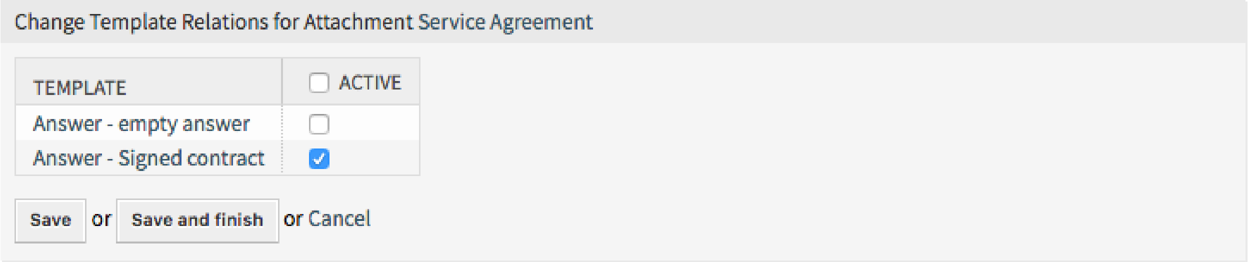
1. Click on an attachment in the *Attachments* column.
2. Select the templates you would like to assign the attachment to.
3. Click on the *Save* or *Save and finish* button.



Change Attachment Relations for Template Answer - Signed contract

ATTACHMENT	□ ACTIVE
Product Manual Annex (troubleshooting-guide.doc)	□
Service Agreement (service-agreement.pdf)	☑

Fig. 52: Change Attachment Relations for Template



Change Template Relations for Attachment Service Agreement

TEMPLATE	□ ACTIVE
Answer - empty answer	□
Answer - Signed contract	☑

Fig. 53: Change Template Relations for Attachment

Note: If several templates or attachments are added to the system, use the filter box to find a particular template or attachment by just typing the name to filter.

Multiple templates or attachments can be assigned in both screens at the same time. Additionally clicking on a template or clicking on an attachment in the relations screen will open the *Edit Template* screen or the *Edit Attachment* screen accordingly.

Warning: Accessing a template or an attachment provides no back link to the relations screen.

2.15 Templates Queues

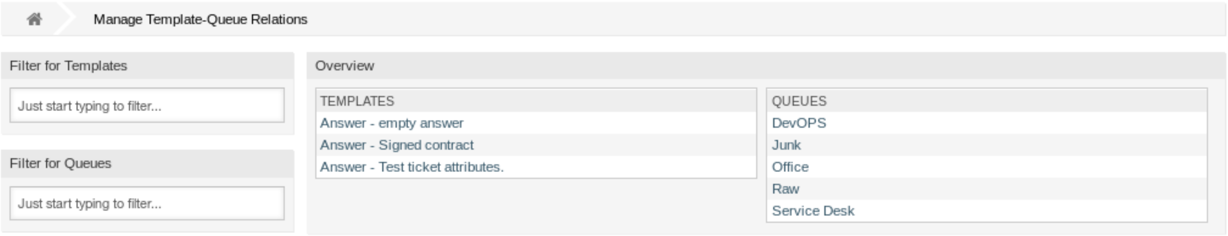
Sharing, distributing and making changes to standard text modules and information for sharing with customers or communication between teams can be an impossible mission.

OTRS can quickly aid you in ensuring that all of your teams have the correct templates available for use by assignment based on the queue.

Use this screen to add one or more templates to one or more queues. To use this function, at least one template and one queue need to have been added to the system. The management screen is available in the *Templates Queues* module of the *Ticket Settings* group.

2.15.1 Manage Templates Queues Relations

To assign a template to a queue:

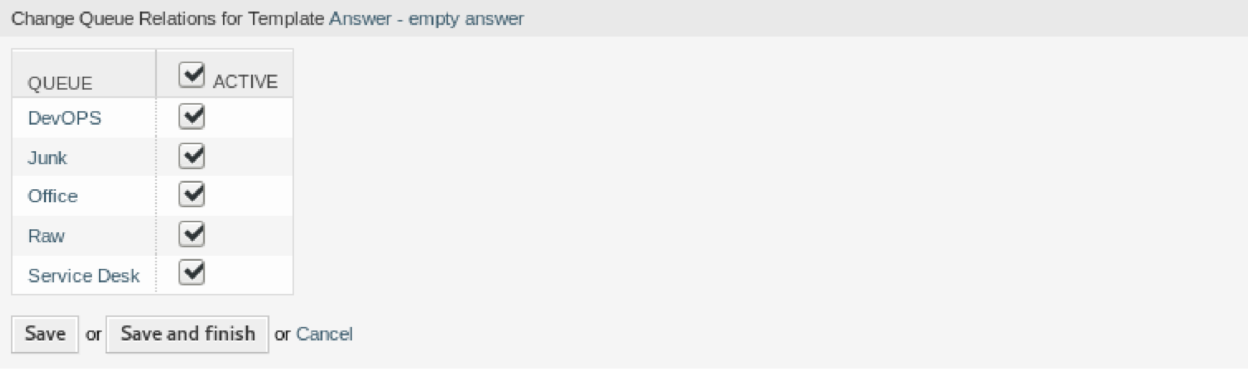

Manage Template-Queue Relations
Filter for Templates
Just start typing to filter...
Filter for Queues
Just start typing to filter...
Overview

TEMPLATES
Answer - empty answer
Answer - Signed contract
Answer - Test ticket attributes.

QUEUES
DevOPS
Junk
Office
Raw
Service Desk

Fig. 54: Manage Template-Queue Relations Screen

1. Click on a template in the *Templates* column.
2. Select the queues you would like to add the template to.
3. Click on the *Save* or *Save and finish* button.

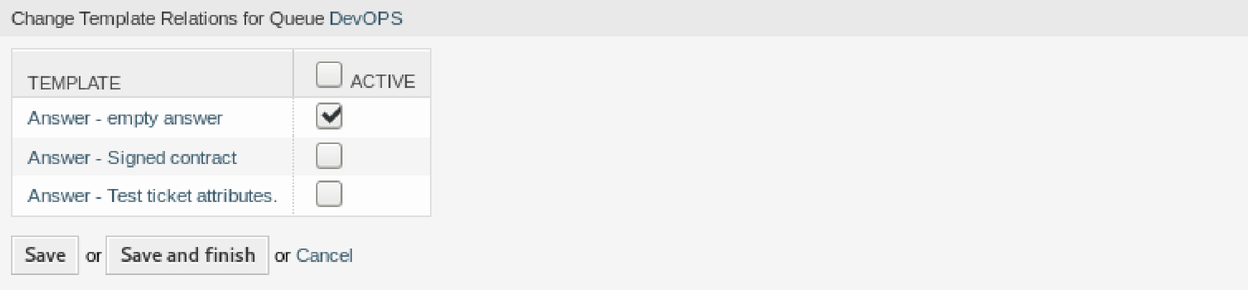

Change Queue Relations for Template Answer - empty answer

QUEUE	☑ ACTIVE
DevOPS	☑
Junk	☑
Office	☑
Raw	☑
Service Desk	☑

Fig. 55: Change Queue Relations for Template

To assign a queue to a template:

1. Click on a queue in the *Queues* column.
2. Select the templates you would like to assign the queue to.
3. Click on the *Save* or *Save and finish* button.


Change Template Relations for Queue DevOPS

TEMPLATE	□ ACTIVE
Answer - empty answer	☑
Answer - Signed contract	□
Answer - Test ticket attributes.	□

Fig. 56: Change Template Relations for Queue

Note: If several templates or queues are added to the system, use the filter box to find a particular template or queue by just typing the name to filter.

Multiple templates or queues can be assigned in both screens at the same time. Additionally clicking on a template or clicking on a queue in the relations screen will open the *Edit Template* screen or the *Edit Queue* screen accordingly.

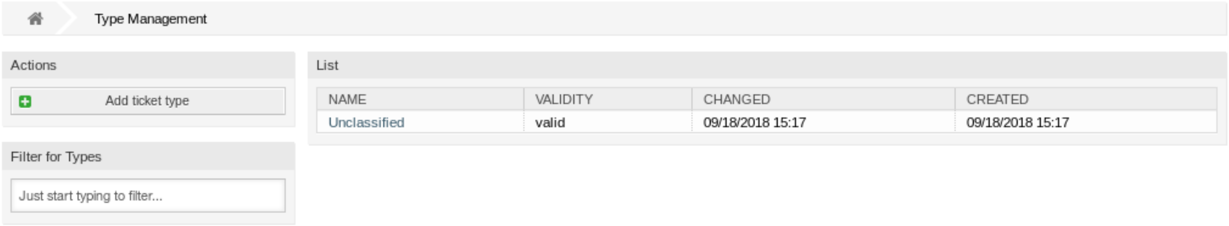
Warning: Accessing a queue or a template provides no back link to the relations screen.

2.16 Types

Good KPIs require knowing the type of work your organization performs. Not all tasks take the same amount of effort even when performed by the same team. Creating a queue structure for this purpose can be overpowered due to the amount of configuration required to create and manage a queue.

OTRS provisions for KPIs with minimal overhead using ticket types. Typical types used in IT service desks are *unclassified*, *incident* and *problem*. You can quickly define new types with ease.

Use this screen to add types to the system. A fresh OTRS installation contains an *unclassified* type by default. The type management screen is available in the *Types* module of the *Ticket Settings* group.



NAME	VALIDITY	CHANGED	CREATED
Unclassified	valid	09/18/2018 15:17	09/18/2018 15:17

Fig. 57: Type Management Screen

Warning: Services must first be activated via System Configuration under the *Administration* group to be selectable in the ticket screens. You may click on the link in the warning message to directly jump to the configuration setting.

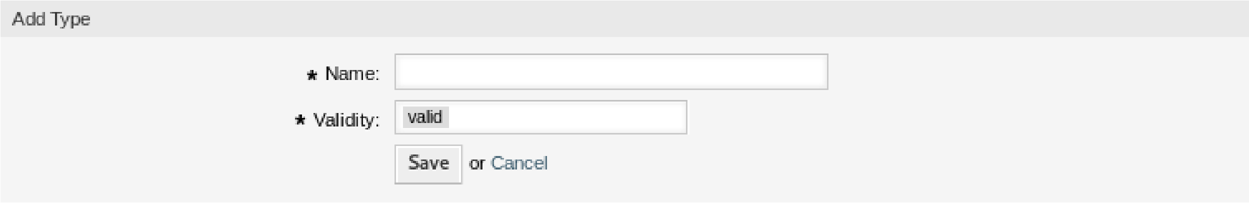
Please activate Type first! →

Fig. 58: Type Activation Warning

2.16.1 Manage Types

To add a type:

1. Click on the *Add Ticket Type* button in the left sidebar.
2. Fill in the required fields.
3. Click on the *Save* button.


Add Type

* Name:

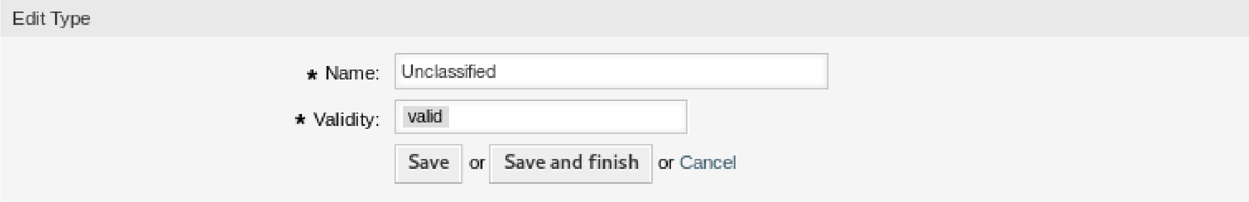
* Validity: valid

Fig. 59: Add Type Screen

Warning: Types can not be deleted from the system. They can only be deactivated by setting the *Validity* option to *invalid* or *invalid-temporarily*.

To edit a type:

1. Click on a type in the list of types.
2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.


Edit Type

* Name: Unclassified

* Validity: valid

Fig. 60: Edit Type Screen

Note: If several types are added to the system, use the filter box to find a particular type by just typing the name to filter.

2.16.2 Type Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Validity * Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.

CHAPTER 3

Communication & Notifications

Clear, complete, and transparent communication is key to ensuring that your organization offers outstanding service to your customers. Your organization would like to give your customers the most consistent service and ensure that your service is highly recognizable, every time. Your teams want to have an identity which instills a specific sense of security and familiarity.

OTRS provides you with the tools to build teams, trust and security to better and uniformly serve your customer. With system addresses you can assign your inbound mails to certain teams and allow them all to use the same address without causing confusion by using team mailboxes via a mail client. Templates, automatic answers, and attachment management let you leverage central management of the most important communications. Many other tools are also available and covered in the sections below.

3.1 Admin Notification

Corporations may need to make general announcements to everyone or publish news to several groups of agents or individuals. Also, OTRS administrator may need to contact with specific agents regarding an event.

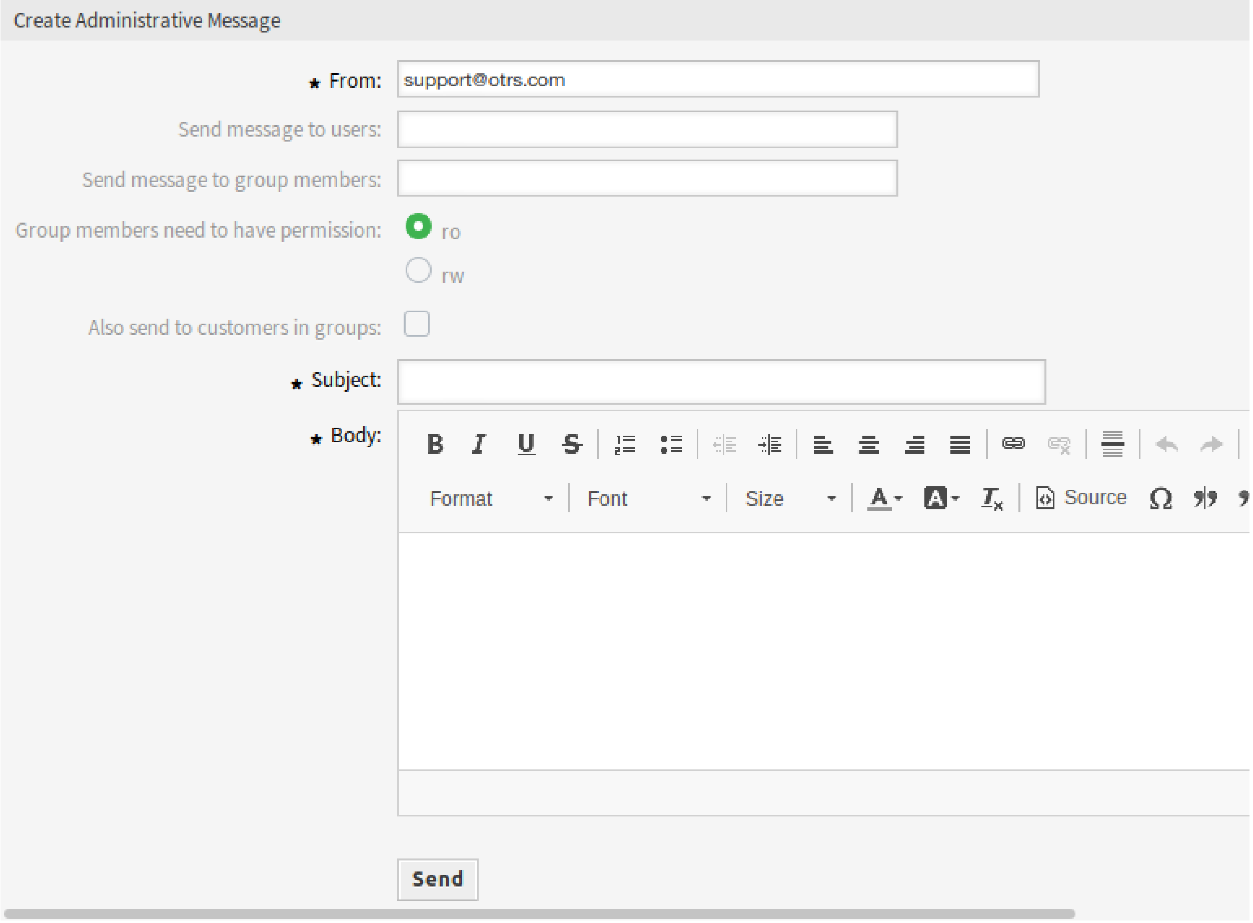
OTRS gives the administration the tool *Admin Notification* making sending announcements and news to the masses of users accurate and timely, to target group of people, simple. Administrators can send notifications based on a specific list of recipients, or a group of users inside OTRS, with powerful text editor enhanced content.

Use this screen to send administrative messages to specific agents, group or role members. The message compose screen is available in the *Admin Notification* module of the *Communication & Notifications* group.

3.1.1 Administrative Message Settings

The following settings are available when composing an administrative message. The fields marked with an asterisk are mandatory.

From * This email address will be added into the *From* field of the message.



Create Administrative Message

★ From:	support@otrs.com
Send message to users:	
Send message to group members:	
Group members need to have permission:	● ro ○ rw
Also send to customers in groups:	□
★ Subject:	
★ Body:	

Fig. 1: Create Administrative Message Screen

Send message to users One or more *Agents* can be selected in this field, to whom the message will be sent.

Send message to group members One or more *Groups* can be selected in this field, to whose members the message will be sent.

Group members need to have permission With these radio buttons can be selected, if a group member need read-only or read-write permissions to receive the message.

Also send to customers in groups Select this checkbox to send the message also for customers in groups.

> **Note:** This option is available only, if the CustomerGroupSupport setting is enabled.

Subject * The subject of the messages.

Body * The body text of the message.

3.2 Appointment Notifications

Missing appointments can damage your image with a customer. Once there is an appointment assigned in the calendar, it' s normal to receive notification:

- Upon a new or changed event
- Upon cancellation of an event
- Before the event, as a reminder

Notification relieves the agent the stress of mentally tracking appointments.

OTRS appointment notifications satisfies this need. Here an administrator can easily set notifications with general rules, including trigger events and filters. Afterward, appointments fitting the bill notify the correct users at the correct time.

Use this screen to add appointment notifications to the system. In a fresh OTRS installation an appointment reminder notification is already added by default. The appointment notification management screen is available in the *Appointment Notifications* module of the *Communication & Notifications* group.

3.2.1 Manage Appointment Notifications

To add an appointment notification:

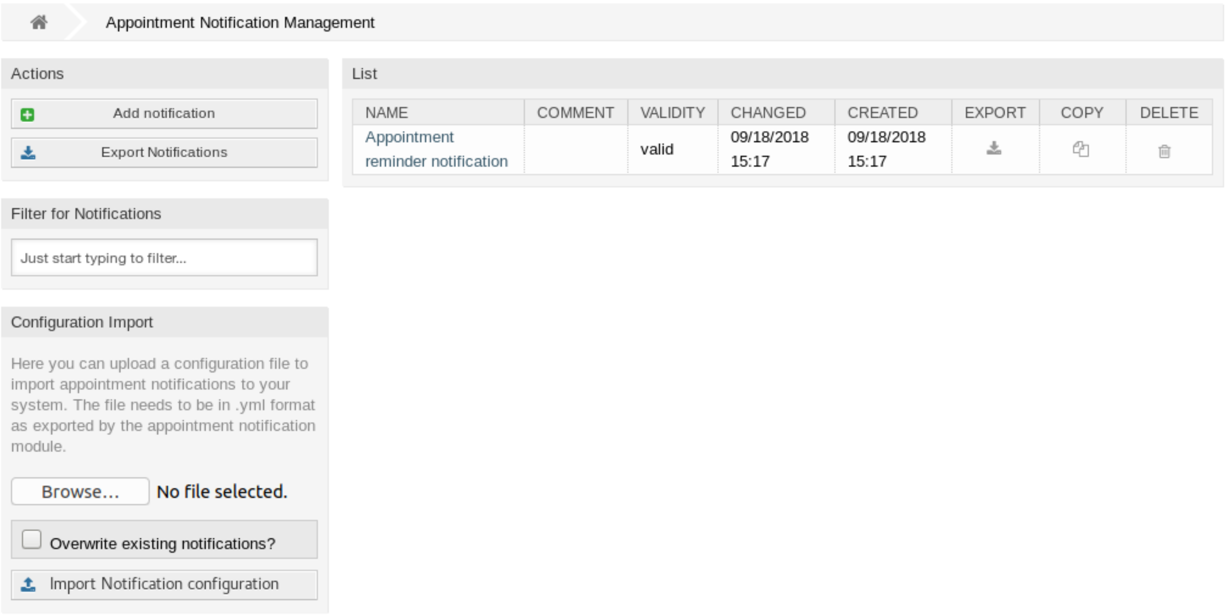
1. Click on the *Add Notification* button in the left sidebar.
2. Fill in the required fields as explained in *Appointment Notification Settings*.
3. Click on the *Save* button.

To edit an appointment notification:

1. Click on an appointment notification in the list of appointment notifications.
2. Modify the fields as explained in *Appointment Notification Settings*.
3. Click on the *Save* or *Save and finish* button.

To delete an appointment notification:

1. Click on the trash icon in the list of appointment notifications.



Appointment Notification Management

Actions

Add notification
Export Notifications

Filter for Notifications

Just start typing to filter...

Configuration Import

Here you can upload a configuration file to import appointment notifications to your system. The file needs to be in .yml format as exported by the appointment notification module.

Browse… No file selected.

□ Overwrite existing notifications?

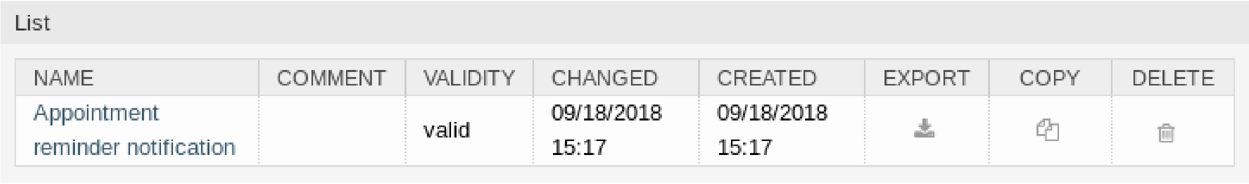
Import Notification configuration

List

NAME	COMMENT	VALIDITY	CHANGED	CREATED	EXPORT	COPY	DELETE
Appointment reminder notification		valid	09/18/2018 15:17	09/18/2018 15:17			

Fig. 2: Appointment Notification Management Screen

2. Click on the *Confirm* button.



List

NAME	COMMENT	VALIDITY	CHANGED	CREATED	EXPORT	COPY	DELETE
Appointment reminder notification		valid	09/18/2018 15:17	09/18/2018 15:17			

Fig. 3: Delete Appointment Notification Screen

To export all appointment notifications:

1. Click on the *Export Notifications* button in the left sidebar.
2. Choose a location in your computer to save the `Export_Notification.yml` file.

To import appointment notifications:

1. Click on the *Browse…* button in the left sidebar.
2. Select a previously exported `.yml` file.
3. Click on the *Overwrite existing notifications?* checkbox, if you would like to overwrite the existing notifications.
4. Click on the *Import Notification configuration* button.

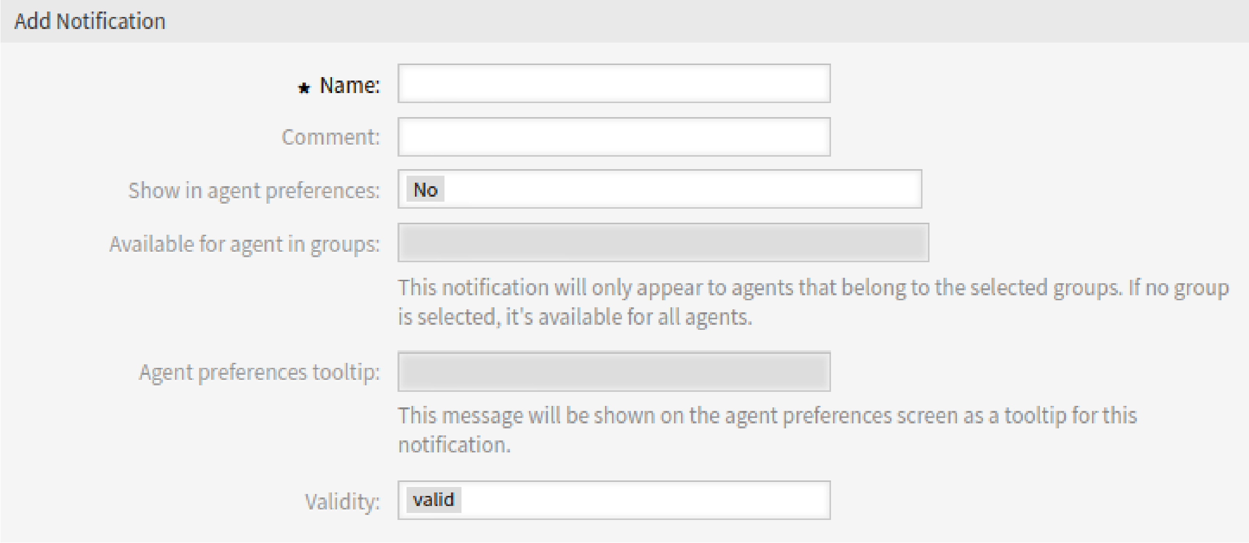
3.2.2 Appointment Notification Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

See also:

For an example, see the default appointment reminder notification which is included in a fresh OTRS installation.

Basic Appointment Notification Settings



Add Notification

⋆ Name:	
Comment:	
Show in agent preferences:	No
Available for agent in groups:	
	This notification will only appear to agents that belong to the selected groups. If no group is selected, it's available for all agents.
Agent preferences tooltip:	
	This message will be shown on the agent preferences screen as a tooltip for this notification.
Validity:	valid

Fig. 4: Appointment Notification Settings - Basic

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

Show in agent preferences Define how the notification should be displayed in agent preferences. The following options are available:

No The notification won't be displayed in agent preferences. The notification is sent to all appropriate agents by the defined method.

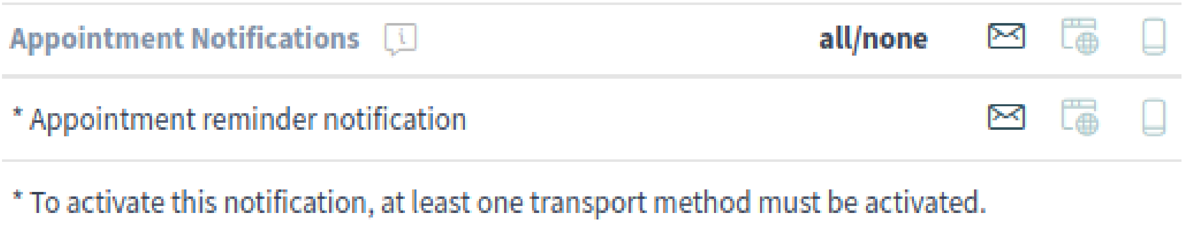
Yes The notification will be displayed in agent preferences for selection. The agents may opt-in or opt-out.

Yes, but require at least one active notification method. The notification will be displayed in agent preferences, but require at least one active notification method. This is annotated by an asterisk next to the name.

Available for agent in groups The notification will only appear to agents that belong to the selected groups. If no group is selected, it is available for all agents.

Agent preferences tooltip This message will be shown on the agent preferences screen as a tooltip for this notification.

Validity * Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.



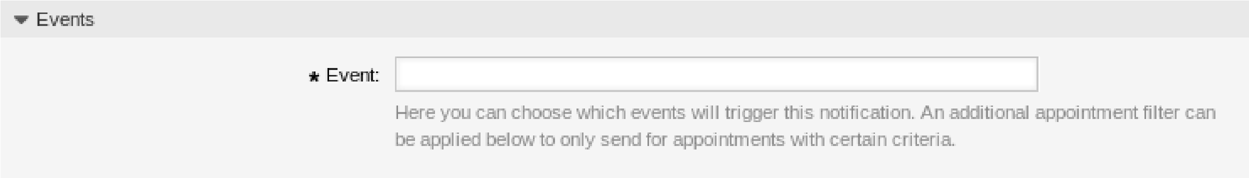
Appointment Notifications all/none

* Appointment reminder notification

* To activate this notification, at least one transport method must be activated.

Fig. 5: Personal Appointment Notification Settings

Appointment Events



* Event:

Here you can choose which events will trigger this notification. An additional appointment filter can be applied below to only send for appointments with certain criteria.

Fig. 6: Appointment Notification Settings - Events

Event Here you can choose which events will trigger this notification. An additional appointment filter can be applied below to only send for appointments with certain criteria.

Possible events are:

`AppointmentCreate` Executed after an appointment has been created.

`AppointmentUpdate` Executed after an appointment has been updated.

`AppointmentDelete` Executed after an appointment has been deleted.

`AppointmentNotification` This is a special appointment event that will be executed by the OTRS daemon in time. If an appointment contains a date/time value for notifications, as already described in this documentation, and such a notification date is reached, the OTRS daemon will execute this kind of event for every related appointment separately.

`CalendarCreate` Executed after a calendar has been created.

`CalendarUpdate` Executed after a calendar has been updated.

Appointment Filter

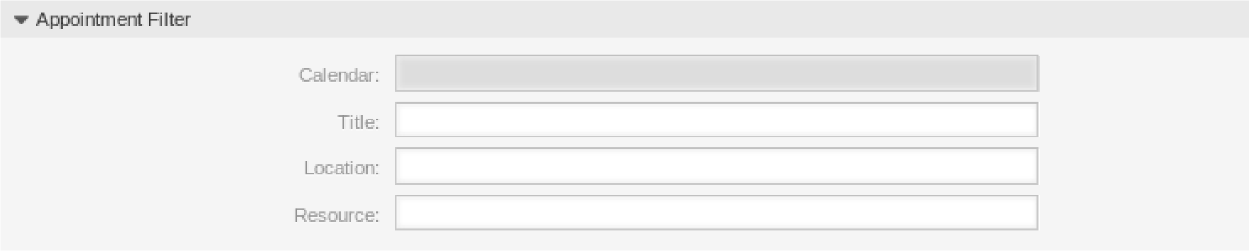
This widget can optionally be used to narrow the list of appointments by matching configured values:

Calendar Select which calendar the related appointment needs to be part of.

Title Filter for a part or complete title of the appointment.

Location Filter for a part or complete location of the appointment.

Resource Choose from a list of teams or resources assigned to the appointments.



▼ Appointment Filter

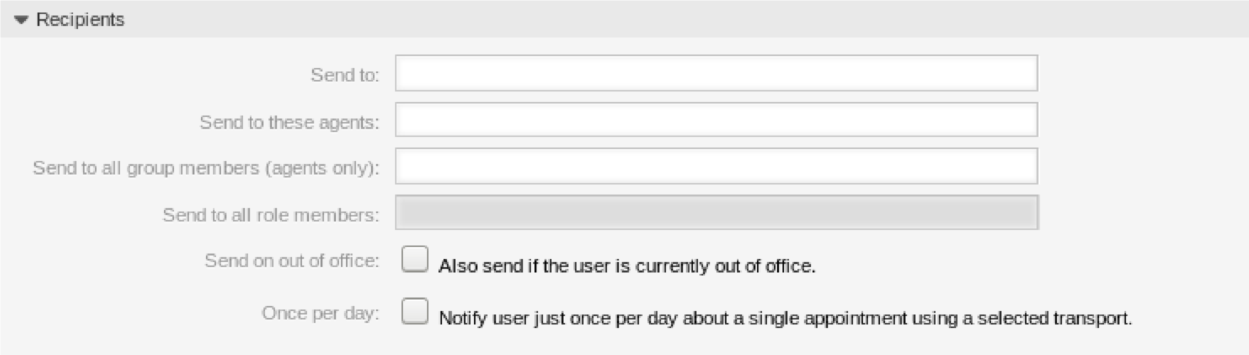
Calendar:

Title:

Location:

Resource:

Fig. 7: Appointment Notification Settings - Appointment Filter



▼ Recipients

Send to:

Send to these agents:

Send to all group members (agents only):

Send to all role members:

Send on out of office: ☐ Also send if the user is currently out of office.

Once per day: ☐ Notify user just once per day about a single appointment using a selected transport.

Fig. 8: Appointment Notification Settings - Recipients

Appointment Notification Recipients

Send to Select which agents should receive the notifications. Possible values are:

- Agent (resources), who are selected within the appointment
- All agents with (at least) read permission for the appointment (calendar)
- All agents with write permission for the appointment (calendar)

Send to these agents One or more agents can be selected who should receive the notifications.

Send to all group members (agents only) One or more groups can be selected whom agents should receive the notifications.

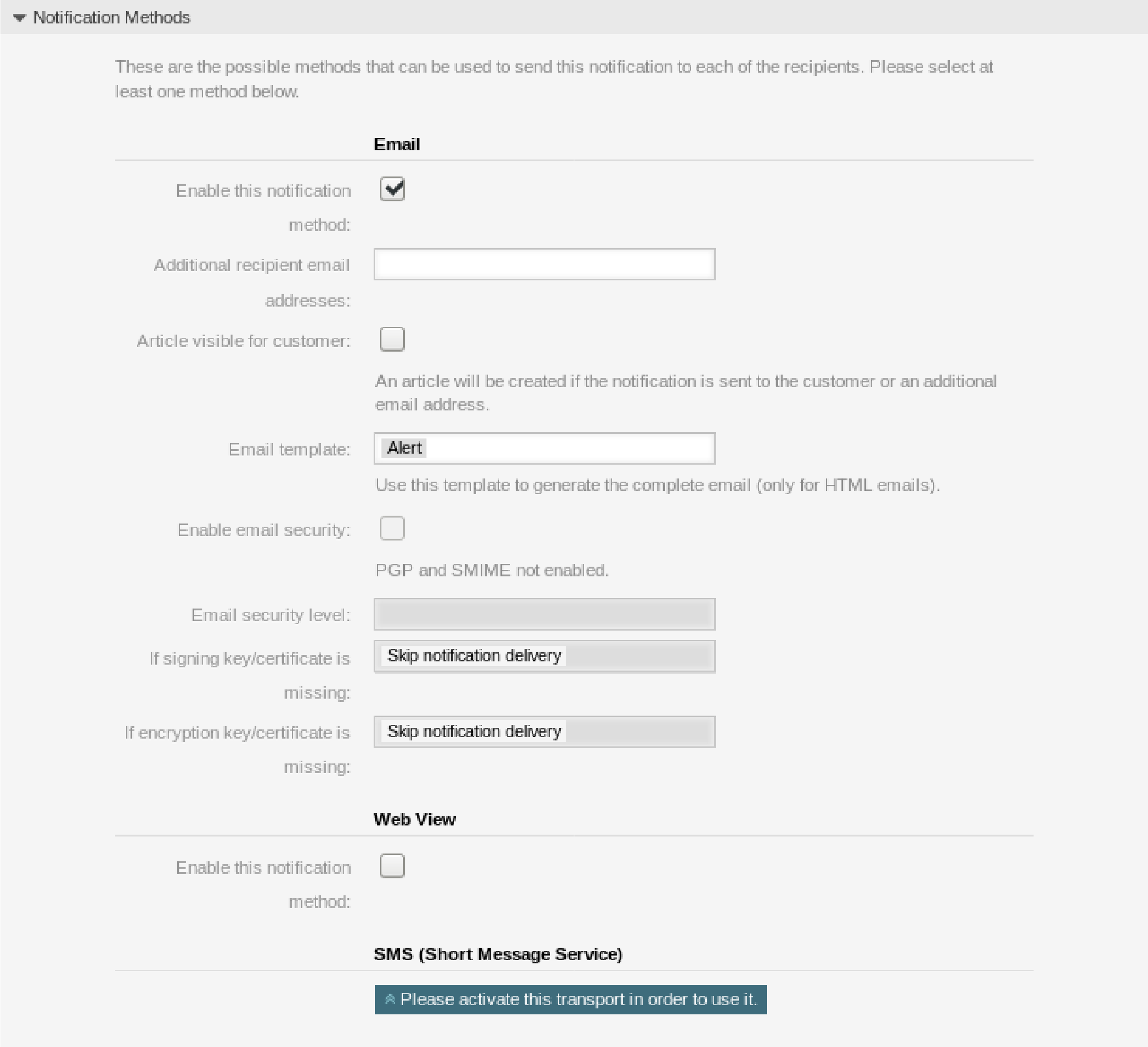
Send to all role members One or more roles can be selected whom agents should receive the notifications.

Send on out of office If this option is checked, the notification will be sent even if the agent is currently out of office.

Once per day Notify users just once per day about a single appointment using a selected transport. If this is the first notification about an appointment, then the notification will be sent. If a notification was already sent before and this option is checked, the OTRS daemon will check the time the last notification was sent. If there was no notification sent in the last 24 hours, the notification will be sent again.

Appointment Notification Methods

Enable this notification method Enable or disable this notification method. A notification method can be email, web view or SMS.



▼ Notification Methods

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

Email

Enable this notification method: ☑

Additional recipient email addresses:

Article visible for customer: □

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

Email template: Alert

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

Enable email security: □

PGP and SMIME not enabled.

Email security level:

If signing key/certificate is missing: Skip notification delivery

If encryption key/certificate is missing: Skip notification delivery

Web View

Enable this notification method: □

SMS (Short Message Service)

Please activate this transport in order to use it.

Fig. 9: Appointment Notification Settings - Notification Methods

Note: To use the SMS notification method, *Cloud Services* need to be enabled.

Additional recipient email addresses Additional recipients can be added here. Use comma or semicolon to separate the email addresses.

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

Email template Select which email template should be used for the notification.

Note: Additional email templates can be added by placing a `.tt` file into the folder `<OTRS_Home>/Kernel/Output/HTML/Templates/Standard/NotificationEvent/Email/`. See the existing email templates for an example.

Enable email security Checking this option will encrypt the notification email.

Note: To use this feature, *PGP Keys* or *S/MIME Certificates* need to be enabled.

Email security level If *Enable email security* is checked, then this setting is activated. The following options are available:

PGP sign only Sign only the notification email with PGP key. If no PGP keys have been added to the system, this option is not visible.

PGP encrypt only Encrypt only the notification email with PGP key. If no PGP keys have been added to the system, this option is not visible.

PGP sign and encrypt Sign and encrypt the notification email with PGP key. If no PGP keys have been added to the system, this option is not visible.

SMIME sign only Sign only the notification email with S/MIME certificate. If no S/MIME certificates have been added to the system, this option is not visible.

SMIME encrypt only Encrypt only the notification email with S/MIME certificate. If no S/MIME certificates have been added to the system, this option is not visible.

SMIME sign and encrypt Sign and encrypt the notification email with S/MIME certificate. If no S/MIME certificates have been added to the system, this option is not visible.

Note: To use this feature, *PGP Keys* or *S/MIME Certificates* need to be enabled.

If signing key/certificate is missing Select the method, that should be used if signing key or certificate is missing.

If encryption key/certificate is missing: Select the method, that should be used if encryption key or certificate is missing.

Appointment Notification Text

The main content of a notification can be added for each languages with localized subject and body text. It is also possible to define static text content mixed with OTRS smart tags.

Subject * The localized subject for a specific language.

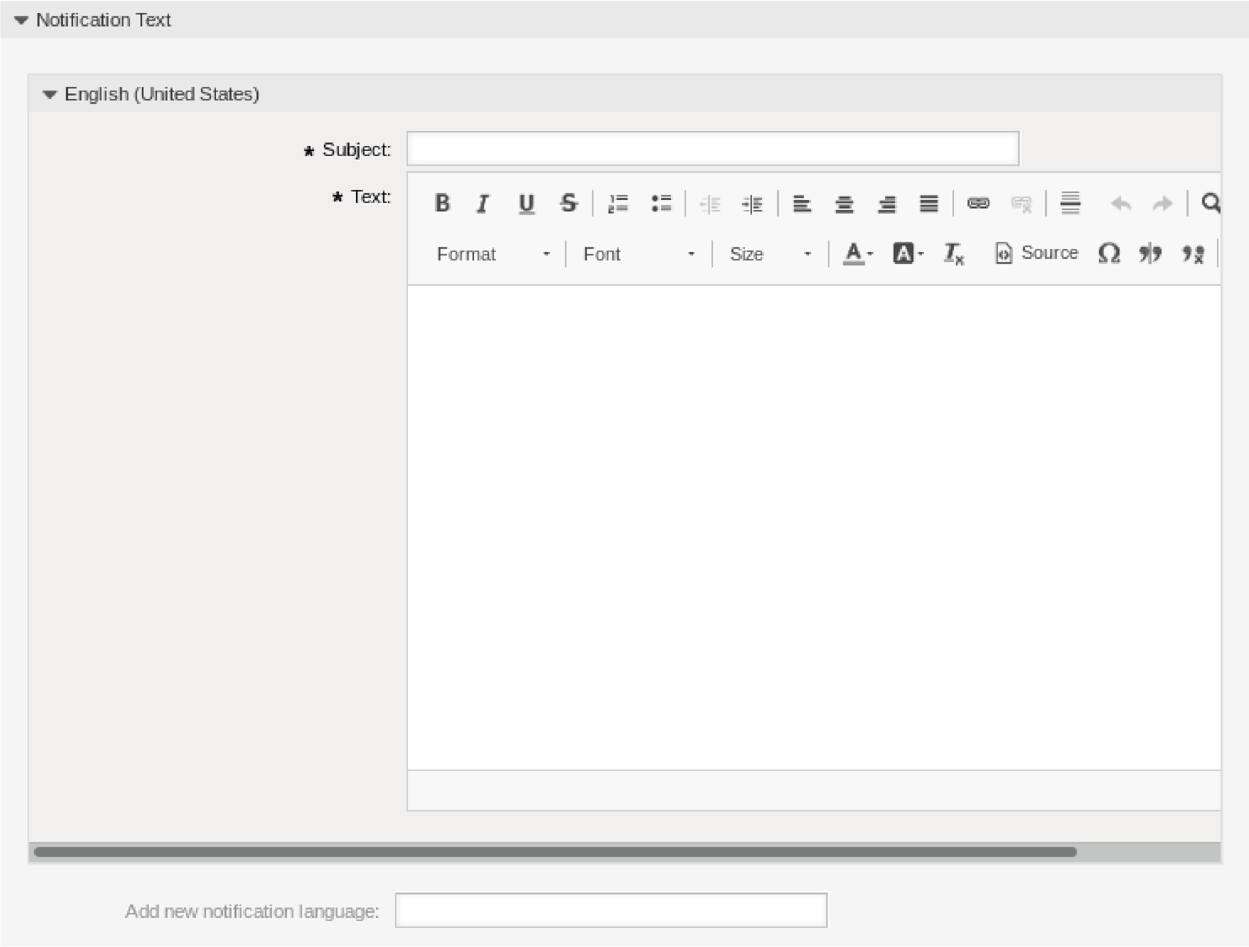


Fig. 10: Appointment Notification Settings - Notification Text

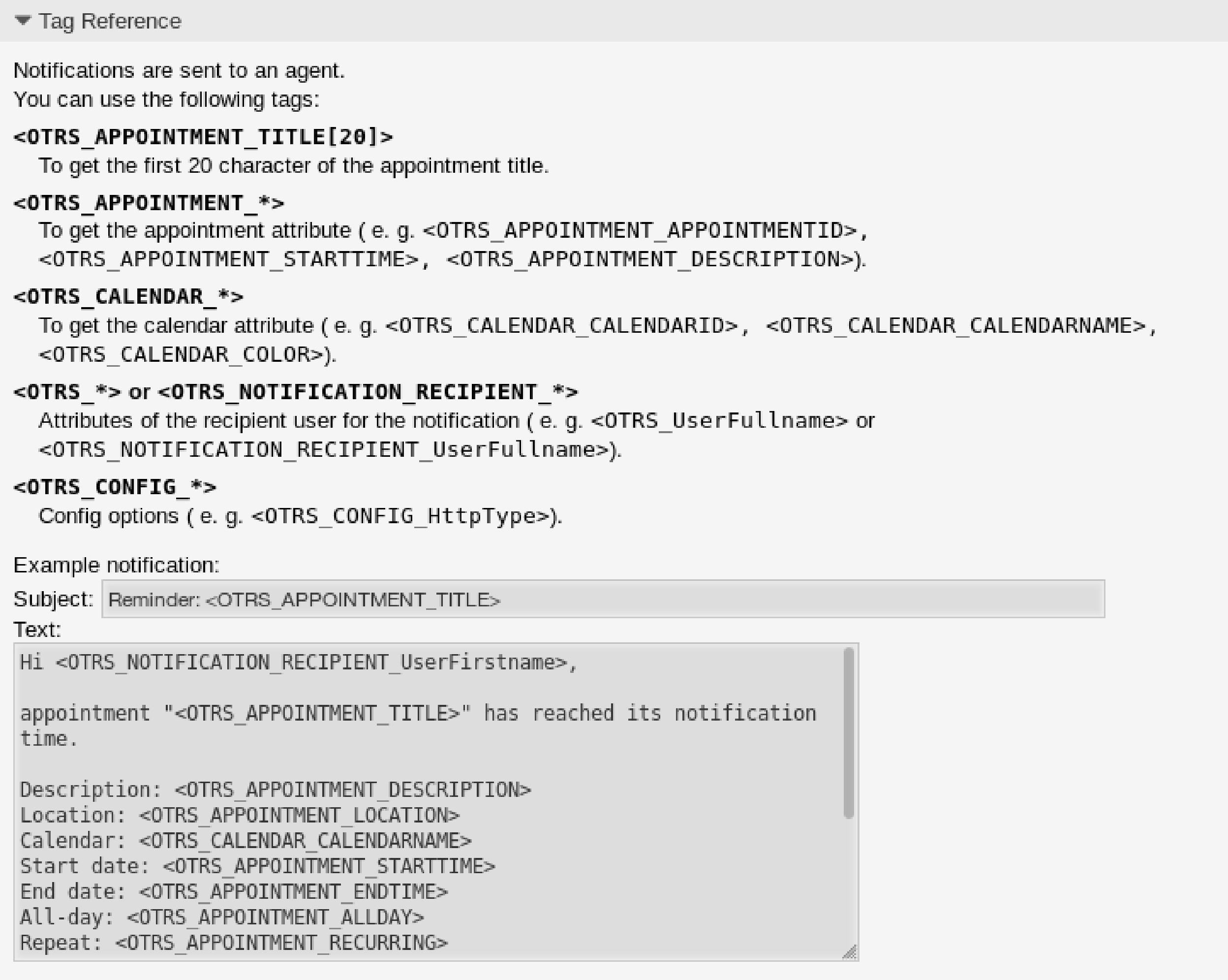
Text * The localized body text for a specific language.

Add new notification language Select which languages should be added to create localized notifications. The language of the customer or agent will be used as found in the customer and agent preferences. Secondarily, the system default language will be chosen. The fall back will always be English.

> **Warning:** Deleting a language from the DefaultUsedLanguages setting that already has a notification text here will make the notification text unusable. If a language is not present or enabled on the system, the corresponding notification text could be deleted if it is not needed anymore.

3.2.3 Appointment Notification Variables

Using variables in the text makes it possible to personalize messages. Variables, known as OTRS tags, are replaced by OTRS when generating the message. Find a list of available tags stems for this resource at the bottom of both add and edit screens.



Notifications are sent to an agent.
You can use the following tags:

<OTRS_APPOINTMENT_TITLE[20]>
To get the first 20 character of the appointment title.

<OTRS_APPOINTMENT_*>
To get the appointment attribute (e. g. <OTRS_APPOINTMENT_APPOINTMENTID>, <OTRS_APPOINTMENT_STARTTIME>, <OTRS_APPOINTMENT_DESCRIPTION>).

<OTRS_CALENDAR_*>
To get the calendar attribute (e. g. <OTRS_CALENDAR_CALENDARID>, <OTRS_CALENDAR_CALENDARNAME>, <OTRS_CALENDAR_COLOR>).

<OTRS_*> or **<OTRS_NOTIFICATION_RECIPIENT_*>**
Attributes of the recipient user for the notification (e. g. <OTRS_UserFullname> or <OTRS_NOTIFICATION_RECIPIENT_UserFullname>).

<OTRS_CONFIG_*>
Config options (e. g. <OTRS_CONFIG_HttpType>).

Example notification:

Subject: Reminder: <OTRS_APPOINTMENT_TITLE>

Text:

```
Hi <OTRS_NOTIFICATION_RECIPIENT_UserFirstname>,

appointment "<OTRS_APPOINTMENT_TITLE>" has reached its notification
time.

Description: <OTRS_APPOINTMENT_DESCRIPTION>
Location: <OTRS_APPOINTMENT_LOCATION>
Calendar: <OTRS_CALENDAR_CALENDARNAME>
Start date: <OTRS_APPOINTMENT_STARTTIME>
End date: <OTRS_APPOINTMENT_ENDTIME>
All-day: <OTRS_APPOINTMENT_ALLDAY>
Repeat: <OTRS_APPOINTMENT_RECURRING>
```

Fig. 11: Appointment Notification Variables

For example, the variable `<OTRS_APPOINTMENT_TITLE[20]>` expands to the first 20 characters of the title allowing a template to include something like the following.

```
Title: <OTRS_APPOINTMENT_TITLE[20]>
```

This tag expands, for example to:

```
Title: Daily meeting in the…
```

The values of the following variables are translated based on the chosen language of the customer user.

```
OTRSTicket_Type
OTRSTicket_State
OTRSTicket_StateType
OTRSTicket_Lock
OTRSTicket_Priority
```

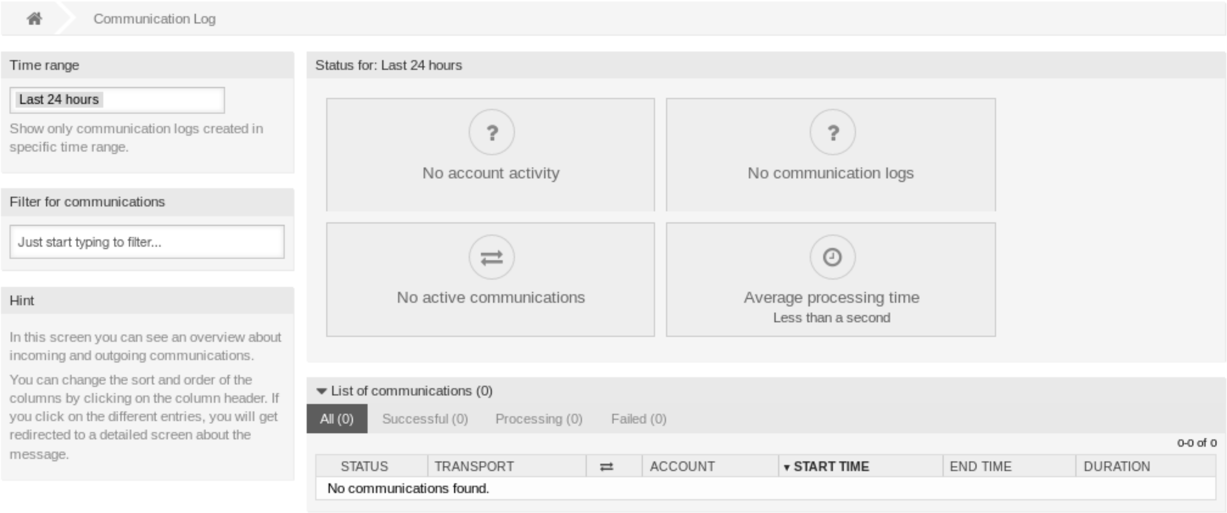
If the language is not supported the default language is applied.

3.3 Communication Log

Managers, leaders, team leads, and system administrators may need to track past communication to follow up specific messages. In some cases, issues arise, and a target recipient did not receive a message. Without access to the mail server logs, tracking the communication is difficult.

OTRS introduces the *Communication Log* module. It' s designed to track the communication: building and spooling the mail and the connection between client and server.

Use this screen to inspect the internal logs about communication handling. The communication log overview screen is available in the *Communication Log* module of the *Communication & Notifications* group.



Time range: Last 24 hours

Show only communication logs created in specific time range.

Just start typing to filter...

In this screen you can see an overview about incoming and outgoing communications.

You can change the sort and order of the columns by clicking on the column header. If you click on the different entries, you will get redirected to a detailed screen about the message.

Average processing time: Less than a second

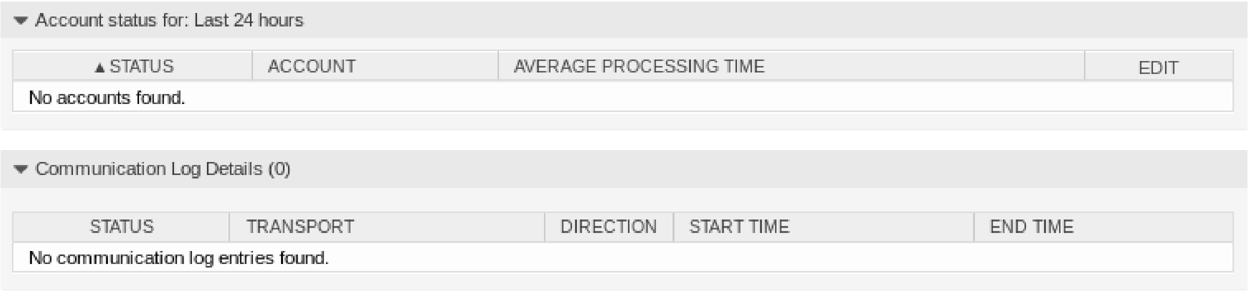
0-0 of 0

STATUS	TRANSPORT	⇄	ACCOUNT	▾ START TIME	END TIME	DURATION
No communications found.						

Fig. 12: Communication Log Overview Screen

3.3.1 Communication Log Overview

The communication log overview page is a dashboard-like screen with several metrics indicating the overall health of the system, depending on filtered communications.



▼ Account status for: Last 24 hours

▲ STATUS	ACCOUNT	AVERAGE PROCESSING TIME	EDIT
No accounts found.			

▼ Communication Log Details (0)

STATUS	TRANSPORT	DIRECTION	START TIME	END TIME
No communication log entries found.				

Fig. 13: Account Status Screen

Account status This widget will signal if you have any issues with configured accounts used for fetching or sending messages.

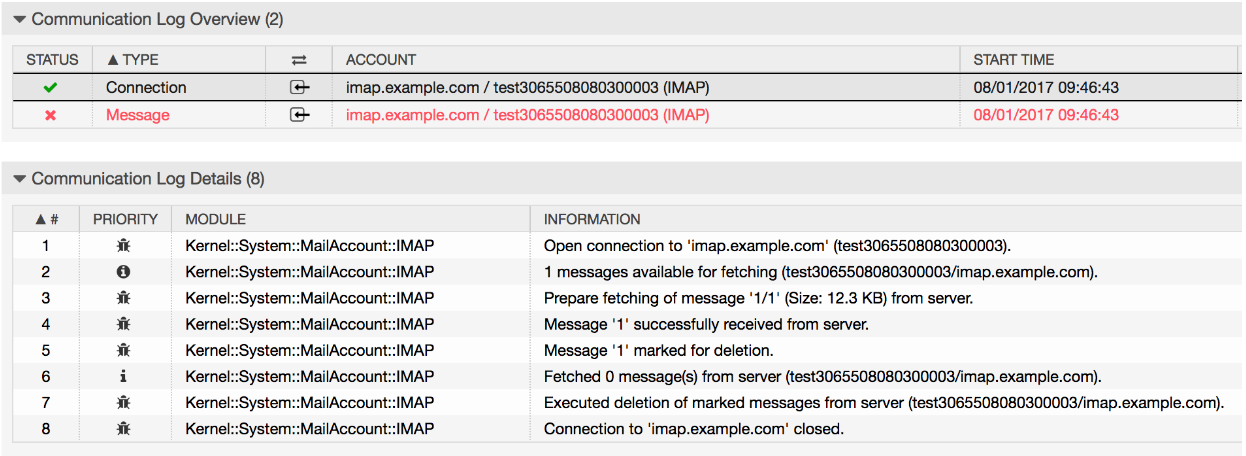
Communication status This widget will notify you if there are any errors with either account connections or message processing.

Communication state This widget will display if there are any active communications currently in the system.

Average processing time This is a cumulative time statistic that is needed to complete a communication.

You can select the time range in the left sidebar in order to filter communications depending on their creation time. In addition to this, you can also dynamically filter for any keywords, state of the communication, and you can sort the overview table by all columns.

If you click on a communication row in any table, you will be presented with a detailed view screen.



▼ Communication Log Overview (2)

STATUS	▲ TYPE	⇄	ACCOUNT	START TIME
✓	Connection	⇥	imap.example.com / test3065508080300003 (IMAP)	08/01/2017 09:46:43
✗	Message	⇥	imap.example.com / test3065508080300003 (IMAP)	08/01/2017 09:46:43

▼ Communication Log Details (8)

▲ #	PRIORITY	MODULE	INFORMATION
1	🐞	Kernel::System::MailAccount::IMAP	Open connection to 'imap.example.com' (test3065508080300003).
2	ℹ	Kernel::System::MailAccount::IMAP	1 messages available for fetching (test3065508080300003/imap.example.com).
3	🐞	Kernel::System::MailAccount::IMAP	Prepare fetching of message '1/1' (Size: 12.3 KB) from server.
4	🐞	Kernel::System::MailAccount::IMAP	Message '1' successfully received from server.
5	🐞	Kernel::System::MailAccount::IMAP	Message '1' marked for deletion.
6	i	Kernel::System::MailAccount::IMAP	Fetched 0 message(s) from server (test3065508080300003/imap.example.com).
7	🐞	Kernel::System::MailAccount::IMAP	Executed deletion of marked messages from server (test3065508080300003/imap.example.com).
8	🐞	Kernel::System::MailAccount::IMAP	Connection to 'imap.example.com' closed.

Fig. 14: Communication Log Detailed View Screen

Every communication can contain one or more logs, which can be of *Connection* or *Message* type.

Connection This type of logs will contain any log messages coming from the modules responsible for connecting to your accounts and fetching/receiving messages.

Message This type of logs will contain any log messages related to specific message processing. Any module working on message themselves can log their actions in this log, giving you a clear overview of what' s going on.

You can filter log entries based on their priority, by choosing desired priority in the left sidebar. Log level rules apply: by selecting a specific priority, you will get log entries that have that priority set and higher, with *Error* being the highest.

3.4 Email Addresses

The main channel of communication with the customers is often email. An organization consists of multiple department or teams. Email addresses differ for each group which is servicing your customer. You may have the following:

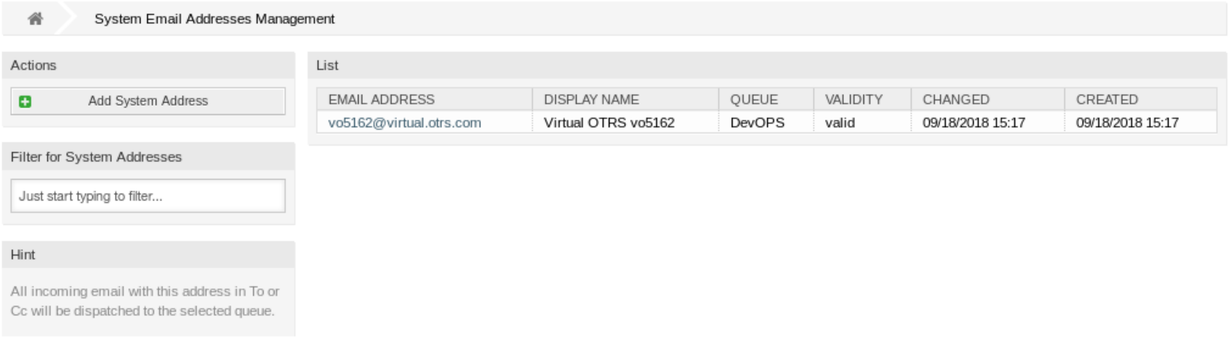
```
support@example.org
hr@exapmle.org
sales@example.org
```

These addresses are just some examples, and you may have many more. You use these channels to receive and send messages, and in mail clients, one can often send with the wrong address.

OTRS manages as many email addresses for your teams as needed. All your email addresses, whether for sending or receiving, are kept and configured nicely in one place. In the *Queue Settings*, the correct address is always chosen preventing that someone working in multiple roles sends an email out with the wrong account.

To enable OTRS to send emails, you need a valid email address to be used by the system. OTRS is capable of working with multiple email addresses, since many support installations need to use more than one. A queue can be linked to many email addresses, and vice versa. The address used for outgoing messages from a queue can be set when the queue is created.

Use this screen to add system email addresses to the system. An email address is already added to the system at installation time of OTRS. The email address management screen is available in the *Email Addresses* module of the *Communication & Notifications* group.


System Email Addresses Management
Actions
Add System Address
Filter for System Addresses
Just start typing to filter...
Hint
List

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

EMAIL ADDRESS	DISPLAY NAME	QUEUE	VALIDITY	CHANGED	CREATED
vo5162@virtual.otrs.com	Virtual OTRS vo5162	DevOPS	valid	09/18/2018 15:17	09/18/2018 15:17

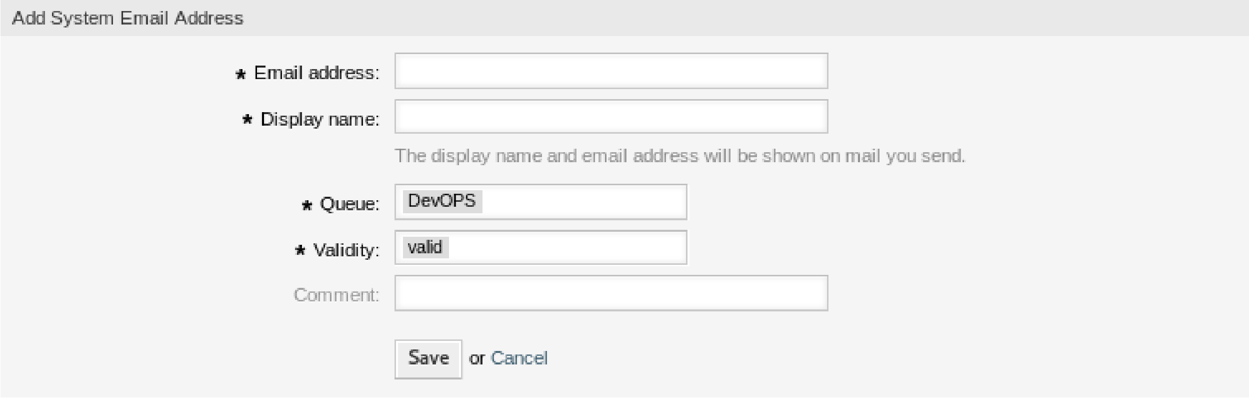
Fig. 15: Email Address Management Screen

3.4.1 Manage Email Addresses

To add an email address:

1. Click on the *Add System Address* button in the left sidebar.

2. Fill in the required fields.
3. Click on the *Save* button.



Add System Email Address

* Email address:	
* Display name:	
	The display name and email address will be shown on mail you send.
* Queue:	DevOPS
* Validity:	valid
Comment:	

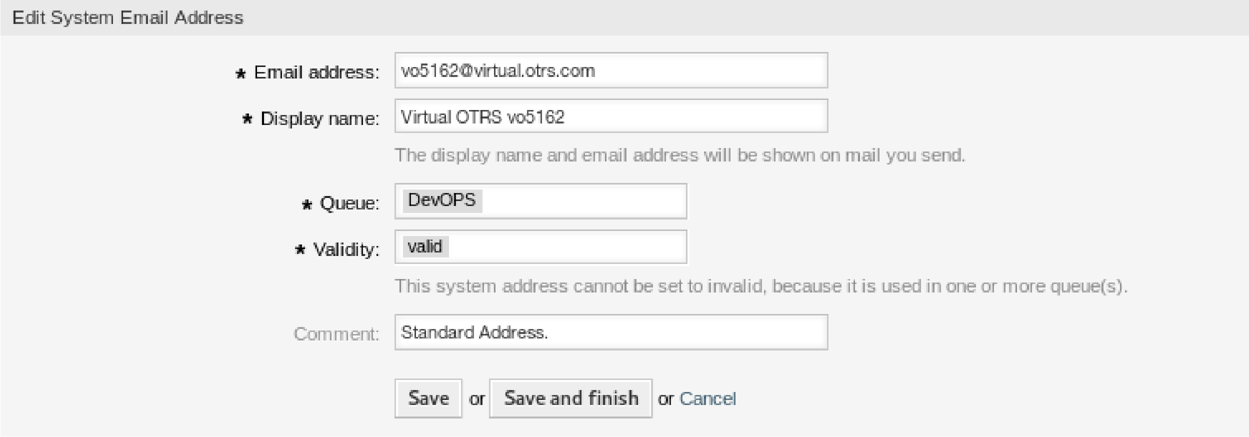
Fig. 16: Email Address Add Screen

Warning: Email addresses can not be deleted from the system. They can only be deactivated by setting the *Validity* option to *invalid* or *invalid-temporarily*.

Note: Once an email address is added and set to valid, OTRS cannot send an email to this address. This prevents loopbacks which could crash your system. If you need to transfer information between departments please use the split article action. This will allow you to create a new ticket to another team for assigning a task, for example.

To edit an email address:

1. Click on an email address in the list of email addresses.
2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.



Edit System Email Address

* Email address:	vo5162@virtual.otrs.com
* Display name:	Virtual OTRS vo5162
	The display name and email address will be shown on mail you send.
* Queue:	DevOPS
* Validity:	valid
	This system address cannot be set to invalid, because it is used in one or more queue(s).
Comment:	Standard Address.

Fig. 17: Email Address Edit Screen

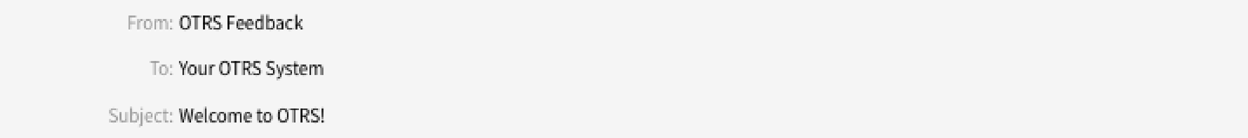
Note: If several email addresses are added to the system, use the filter box to find a particular email address by just typing the name to filter.

3.4.2 Email Address Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Email address * The email address to be added.

Display name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the sender information of the article.



From: OTRS Feedback

To: Your OTRS System

Subject: Welcome to OTRS!

Fig. 18: Sender Information

Queue * The queue, to which the email address will be added as default email address.

Note: This setting will apply if the email is distributed via the recipient address. This setting can be overridden by *Postmaster Filters* or in the *Mail Account Settings* when *Dispatching by selected Queue* is chosen.

Validity * Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.

Note: An email address can only be set to *invalid* or *invalid-temporarily*, if it is not assigned to any queue.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

3.4.3 Setting up Outgoing Emails

For outgoing emails, OTRS can be configured in a variety of ways. The best option to do this might depend on your specific circumstances.

Sending Mails via Local MTA

Using a local mail transfer agent (MTA) to send mails directly or relay mails to a smart host is the **recommended** option. It provides local mail queuing in the case of network connectivity issues and provides you with the capability to authenticate against multiple outgoing mail accounts or accounts using other than basic authentication methods.

A default setup could use the *Sendmail* binary and *Postfix*.

Sending emails via SMTP (Simple Mail Transfer Protocol) relay via a smart host can be used in case you are using another mail server in your network, or on the internet. Your local MTA must be configured with the credentials for authentication and the target MTA must allow the user to send mails as other users of the domain.

Sending emails in the name of another domain requires either a sender policy framework (SPF) record, allowing the local server to send as the user, or in the case that the local mail server should act as the mail server for a target domain, mail exchange resource record (MX) and domain name service (DNS) C-name records for the local server are required. The MTA must, in both cases, be correctly configured to send as the target domain.

Note: An SPF record is not necessarily required. However, if the sender domain has an SPF record set, the MTA’s IP address must be listed in it, otherwise the SPF check will fail. Alternatively, you can send outbound mails to a relay host, and in that case the relay’s IP address must be listed in the SPF record of the sender domain. If the sender domain has no SPF record, there is nothing wrong with the OTRS MTA delivering outbound mails directly.

Sending Mails via External Email Provider

Alternatively, it is also possible to use an external email provider for outgoing emails.

Note: Only regular password authentication is supported. For advanced OAuth2 support and in case you are an OTRS customer, please raise a support ticket in the OTRS customer portal.

Note: In case you are an OTRS customer, you can change the SMTP configuration by raising a support ticket in the OTRS customer portal.

To setup the system to use *Office 365* as email provider:

1. Read the POP, IMAP, and SMTP settings for Outlook.com and Add your Outlook.com account to another mail app or smart device chapters in the official documentation.
2. Copy the one-time application password shown in the administrator area of **Office 365**.
3. Go to the *System Configuration* screen of **OTRS** and add the following settings:
   - `SendmailModule` → `Kernel::System::Email::SMTPTLS`
   - `SendmailModule::AuthPassword` → the application password from Office 365
   - `SendmailModule::AuthUser` → *email-address@company.tld*
   - `SendmailModule::Host` → *SMTP.office365.com*
   - `SendmailModule::Port` → *587*

To setup the system to use *Gmail* as email provider:

1. Read the Send email from a printer, scanner, or app chapter in the official Help Center.
2. Go to the *System Configuration* screen of **OTRS** and add the following settings:
   - `SendmailModule` → `Kernel::System::Email::SMTPTLS`
   - `SendmailModule::AuthPassword` → your email password
   - `SendmailModule::AuthUser` → *email-address@company.tld*

- `SendmailModule::Host` → *smtp-relay.gmail.com*
- `SendmailModule::Port` → *587*

To setup the system to use another external email provider:

1. Read the official documentation of the external provider.
2. Go to the *System Configuration* screen of **OTRS** and add the following settings:
   - `SendmailModule` → `Kernel::System::Email::SMTPTLS`
   - `SendmailModule::AuthPassword` → your email password
   - `SendmailModule::AuthUser` → your email address or username
   - `SendmailModule::Host` → the SMTP address of the external provider
   - `SendmailModule::Port` → the port for the SMTP server

See also:

You can find detailed instructions in the *PostMaster Mail Accounts* chapter about how to configure incoming emails.

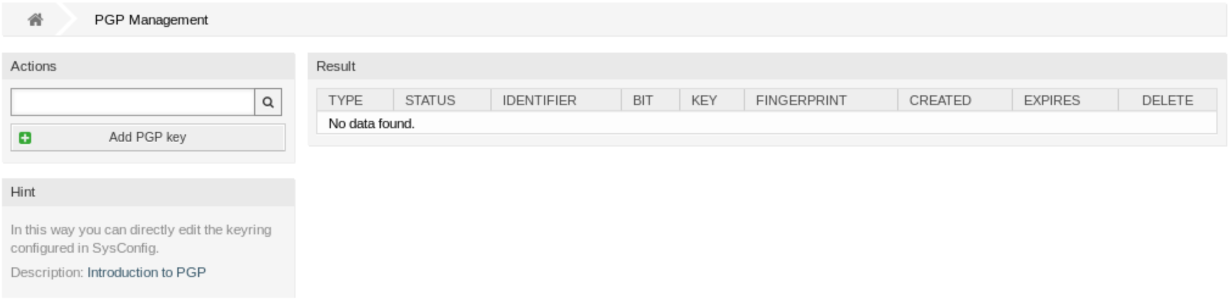
3.5 PGP Keys

Secure communications protect your customers and you. In the GDPR encryption is explicitly mentioned as one of the security and personal data protection measures in a few articles. Although under the GDPR encryption is not mandatory, it is indeed essential in some areas.

OTRS empowers you to encrypt communications where needed by means of *S/MIME Certificates* or *PGP Keys*.

Note: Setup of services and software required for encryption are not covered here because of independence to this software.

Use this screen to add PGP keys to the system. The PGP management screen is available in the *PGP Keys* module of the *Communication & Notifications* group.



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

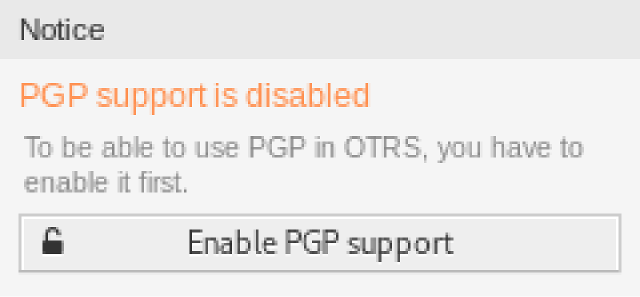
In this way you can directly edit the keyring configured in SysConfig.

Description: Introduction to PGP

Fig. 19: PGP Management Screen

3.5.1 Manage PGP Keys

Note: To be able to use PGP keys in OTRS, you have to activate its setting first.



Notice

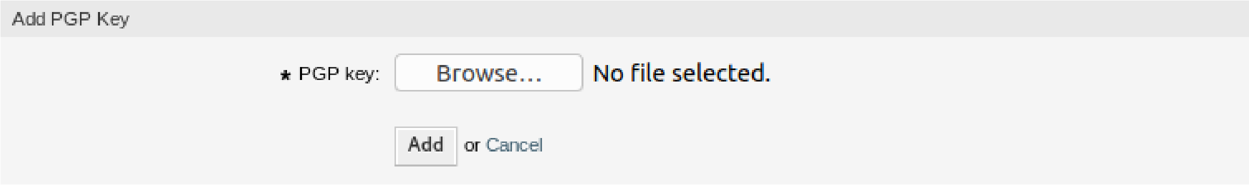
PGP support is disabled

To be able to use PGP in OTRS, you have to enable it first.

Fig. 20: Enable PGP Support

To add a PGP key:

1. Click on the *Add PGP Key* button in the left sidebar.
2. Click on *Browse*··· button to open the file dialog.
3. Select a PGP key from the file system.
4. Click on the *Add* button.



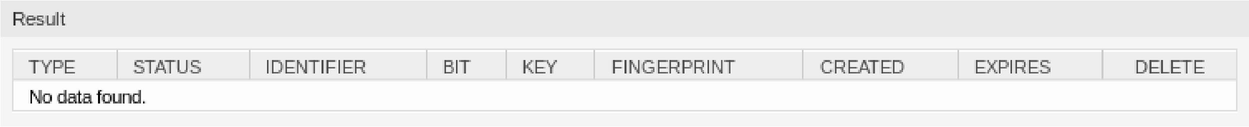
Add PGP Key

∗ PGP key: Browse… No file selected.

Fig. 21: Add PGP Key Screen

To delete a PGP key:

1. Click on the trash icon in the list of PGP keys.
2. Click on the *Confirm* button.



Result

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

Fig. 22: Delete PGP Key Screen

Note: If several PGP keys are added to the system, use the search box to find a particular PGP key.

To adjust the PGP settings of the system:

1. Go to *System Configuration* screen.
2. Navigate to *Core* → *Crypt* → *PGP* in the navigation tree.

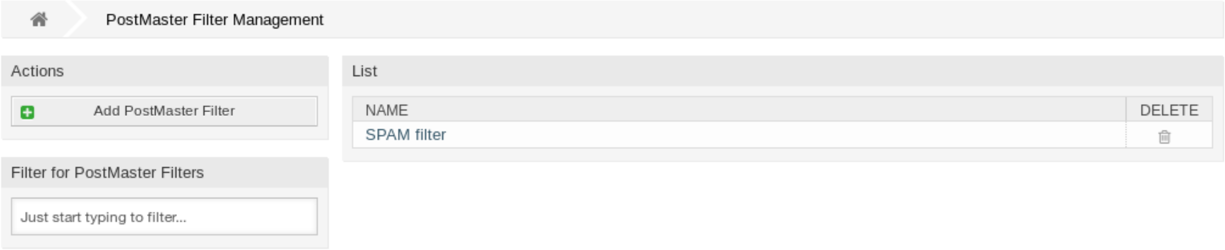
3. Review the settings.

3.6 Postmaster Filters

Pre-sorting standard mail done in a mail room takes care that not every piece of mail sent to the office goes to the same group of people. After a second look at the envelope, rerouting occurs where needed.

OTRS uses so-called *postmaster filters* to read the email' s envelope and take further action. Depending upon, for example, a subject or sender, an email bound for the service desk could land in a sub-queue or be redirected to a completely different team to create transparency and give your customer the fastest service possible.

Use this screen to add postmaster filters to the system. The postmaster filter management screen is available in the *PostMaster Filters* module of the *Communication & Notifications* group.


PostMaster Filter Management
Actions
Add PostMaster Filter
Filter for PostMaster Filters
Just start typing to filter...
List

NAME	DELETE
SPAM filter	🗑

Fig. 23: Postmaster Filter Management Screen

3.6.1 Manage Postmaster Filters

Note: When adding or editing a postmaster filter, please keep in mind that they are evaluated in *ASCIIbetical* order by name.

To add a postmaster filter:

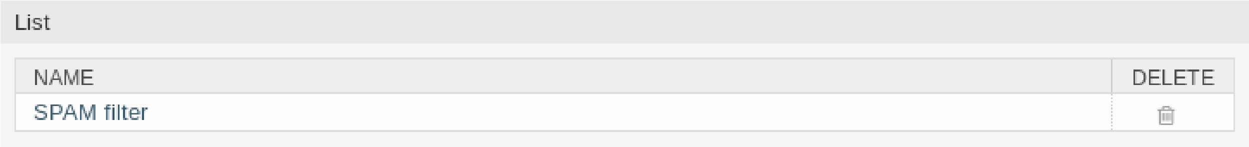
1. Click on the *Add PostMaster Filter* button in the left sidebar.
2. Fill in the required fields.
3. Click on the *Save* button.

To edit a postmaster filter:

1. Click on a postmaster filter in the list of postmaster filters.
2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.

To delete a postmaster filter:

1. Click on the trash icon in the list of postmaster filters.
2. Click on the *Confirm* button.



List

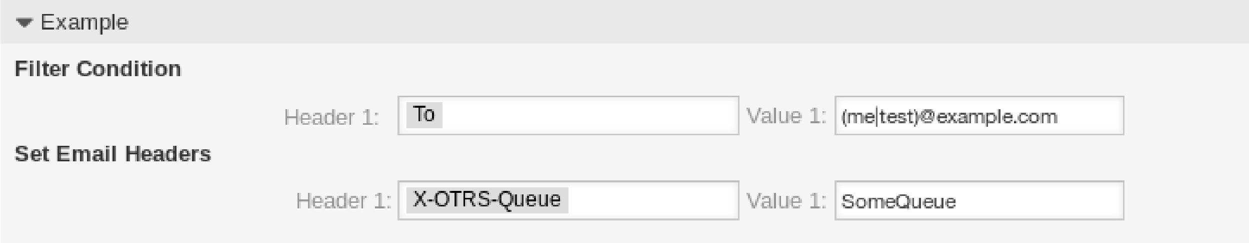
NAME	DELETE
SPAM filter	🗑

Fig. 24: Delete Postmaster Filter Screen

Note: If several postmaster filters are added to the system, a filter box is useful to find a particular postmaster filter by just typing to filter.

3.6.2 Postmaster Filter Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.



▾ Example

Filter Condition

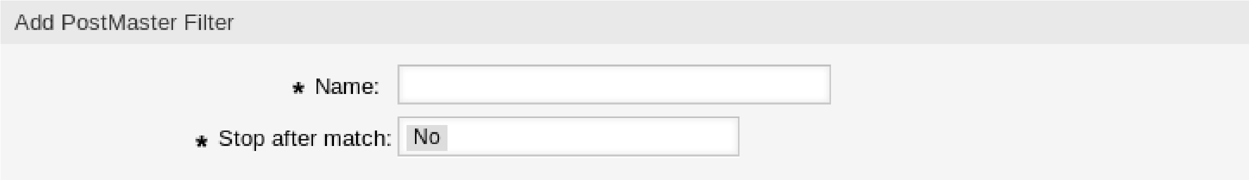
Header 1: To Value 1: (me|test)@example.com

Set Email Headers

Header 1: X-OTRS-Queue Value 1: SomeQueue

Fig. 25: Postmaster Filter Settings Example

Basic Postmaster Filter Settings



Add PostMaster Filter

* Name:

* Stop after match: No

Fig. 26: Postmaster Filter Settings - Basic

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

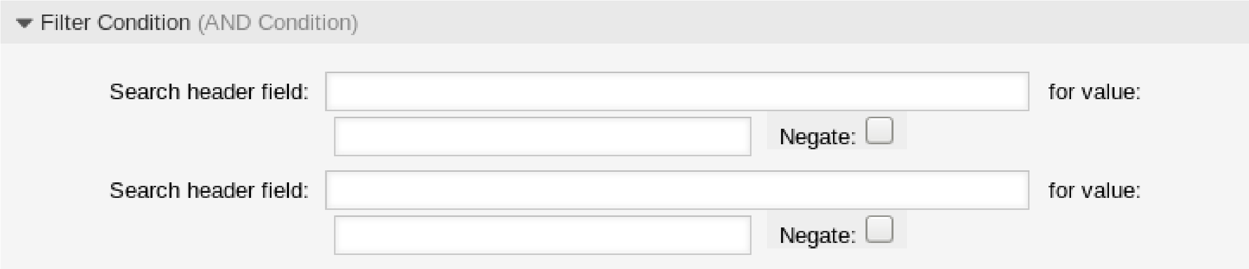
> **Note:** When adding or editing one of the postmaster filters, remember multiple filters may apply to a single mail. Rules are executed and sorted by the ASCII value of the names. Based on the sorted order in the overview, they are applied from top to bottom. Look at the ASCII table to see how to sort your names based on the *ASCIIbetical* order.

Stop after match * Postmaster filters are evaluated in *ASCIIbetical* order. This setting defines the evaluation of the subsequent postmaster filters.

No All postmaster filters are executed.

Yes The current postmaster filter is still evaluated, but evaluation of the remaining filters is canceled.

Filter Condition



▾ Filter Condition (AND Condition)

Search header field: [ ] for value: [ ] Negate: □

Search header field: [ ] for value: [ ] Negate: □

Fig. 27: Postmaster Filter Settings - Filter Condition

A postmaster filter consists of one or more conditions that must be met in order for the defined actions to be executed on the email. Filter conditions can be defined for specific mail header entries or for strings in the mail body.

Search header field …for value Select a mail header or an `X-OTRS` header from the first drop-down list, and enter a value as search term for the selected mail header to the second field. Even regular expressions can be used for extended pattern matching.

A list of mail header entries can be found in RFC5322. It is also possible to define `X-OTRS` headers as filter condition. The different `X-OTRS` headers and their meaning are the following:

`X-OTRS-AttachmentCount` This contains as value the number of attachments which are contained in the email (e.g. *0* for mails without attachments).

`X-OTRS-AttachmentExists` Depending on whether attachments are included in the email this `X-OTRS` header is set to `yes`, or it has a `no` value if no attachments are included.

`X-OTRS-BodyDecrypted` If the incoming mail was encrypted, it is possible to add a search term to look for the body of the incoming encrypted mail.

`X-OTRS-CustomerNo` Set the customer ID for the ticket.

`X-OTRS-CustomerUser` Set the customer user for the ticket.

`X-OTRS-DynamicField-<DynamicFieldName>` Saves an additional information value for the ticket on *<DynamicFieldName>* dynamic field. The possible values depend on dynamic field configuration (e.g. text: *Notebook*, date: *2010-11-20 00:00:00*, integer: *1*).

`X-OTRS-FollowUp-*` These headers are the same as the ones without the `FollowUp` prefix, but these headers are applied only for follow-up mails.

`X-OTRS-FollowUp-State-Keep` If set to *1*, the incoming follow-up message will not change the ticket state. For this purpose the header can be customized in the system configuration using option `KeepStateHeader`.

`X-OTRS-Ignore` If set to `Yes` or `True`, the incoming message will completely be ignored and never delivered to the system.

`X-OTRS-IsVisibleForCustomer` Controls if the article is shown to customer users. Possible values are *0* or *1*.

`X-OTRS-Lock` Set the lock state of a ticket. Possible values are `lock` or `unlock`.

`X-OTRS-Loop` If set to `Yes` or `True`, no auto answer is delivered to the sender of the message (mail loop protection).

X-OTRS-Owner Set the agent as owner for the ticket.

X-OTRS-OwnerID Set the agent ID as owner for the ticket.

X-OTRS-Priority Set the priority for the ticket.

X-OTRS-Queue Defines the queue in which the ticket should be sorted. If a queue is set with this header, this setting has priority over all other filter rules that refer to queues. If you use a sub-queue, specify it as *Parent::Sub*.

X-OTRS-Responsible Set the agent as responsible for the ticket.

X-OTRS-ResponsibleID Set the agent ID as responsible for the ticket.

X-OTRS-SenderType Set the sender type for the ticket. Possible values are `agent`, `system` or `customer`.

X-OTRS-Service Set the service for the ticket. If you use a sub-service, specify it as *Parent::Sub*.

X-OTRS-SLA Set the service level agreement for the ticket.

X-OTRS-State Set the state for the ticket.

X-OTRS-State-PendingTime Set the pending time for the ticket (you should sent a pending state via `X-OTRS-State`). You can specify absolute dates like *2010-11-20 00:00:00* or relative dates, based on the arrival time of the email. Use the form `+ $Number $Unit`, where `$Unit` can be *s* (seconds), *m* (minutes), *h* (hours) or *d* (days). Only one unit can be specified. Examples of valid settings: *+50s* (pending in 50 seconds), *+30m* (30 minutes), *+12d* (12 days).

Note: Settings like *+1d 12h* are not possible. You can specify *+36h* instead.

X-OTRS-Title Set the title for the ticket.

X-OTRS-Type Set the type for the ticket.

Warning: These headers must be manually injected into the mail by means not provided for by OTRS. OTRS only accepts `X-OTRS` headers from trusted sources.

See also:

The *Mail Account Settings* defines the trust level.

Negate If checked, the condition will use the negate search term.

Set Email Headers

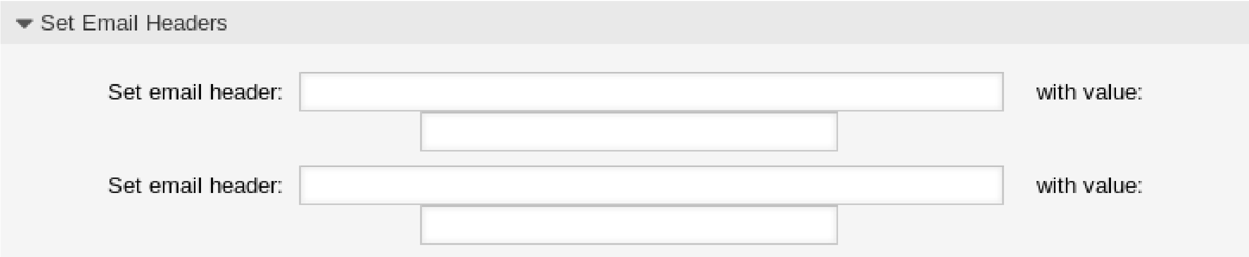


Fig. 28: Postmaster Filter Settings - Set Email Headers

In this section you can choose the actions that are triggered if the filter rules match.

Set email header …with value Select an `X-OTRS` header from the first drop-down list, and add a value to the second field that should be set as value of the selected `X-OTRS` header.

See also:

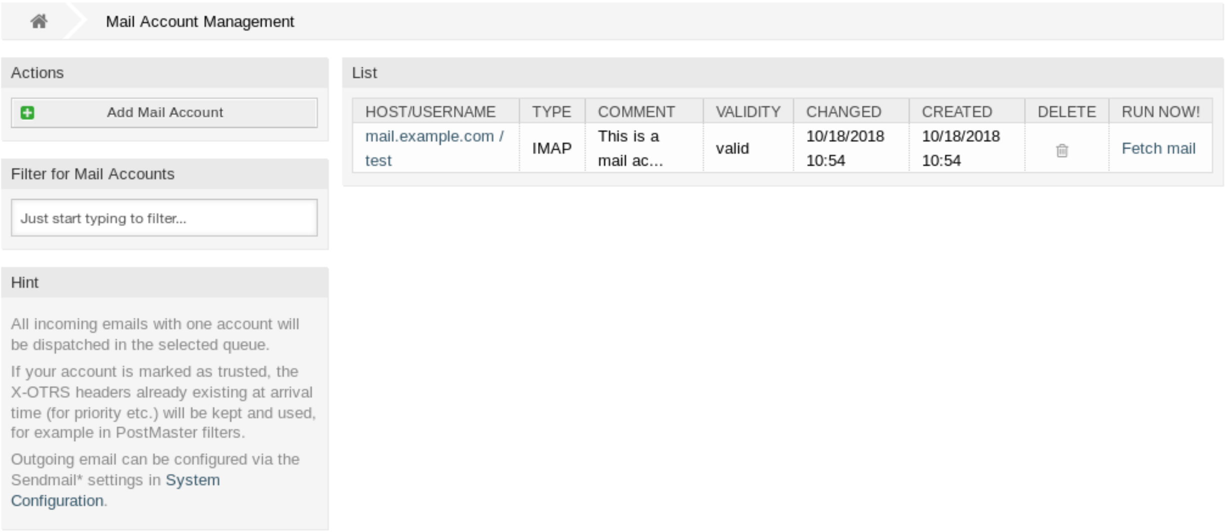
The `X-OTRS` headers are already described above.

3.7 PostMaster Mail Accounts

Just as a company doesn't just have one department which receives traditional mail, your service desk will also serve multiple teams. Each team can have its physical email mailbox.

OTRS eases setup for email mailboxes. OTRS manages polling one or multiple email mailboxes of any internet standard type.

Use this screen to add mail accounts to the system. The mail account management screen is available in the *PostMaster Mail Accounts* module of the *Communication & Notifications* group.



HOST/USERNAME	TYPE	COMMENT	VALIDITY	CHANGED	CREATED	DELETE	RUN NOW!
mail.example.com / test	IMAP	This is a mail ac...	valid	10/18/2018 10:54	10/18/2018 10:54	🗑	Fetch mail

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

If your account is marked as trusted, the X-OTRS headers already existing at arrival time (for priority etc.) will be kept and used, for example in PostMaster filters.

Outgoing email can be configured via the Sendmail* settings in System Configuration.

Fig. 29: Mail Account Management Screen

Warning: When fetching mail, OTRS deletes the mail from the POP or IMAP server. There is no option to also keep a copy on the server. If you want to retain a copy on the server, you should create forwarding rules on your mail server. Please consult your mail server documentation for details.

Note: If you choose IMAP, you can specify a folder for collection. Selective dispatching of mails is then possible.

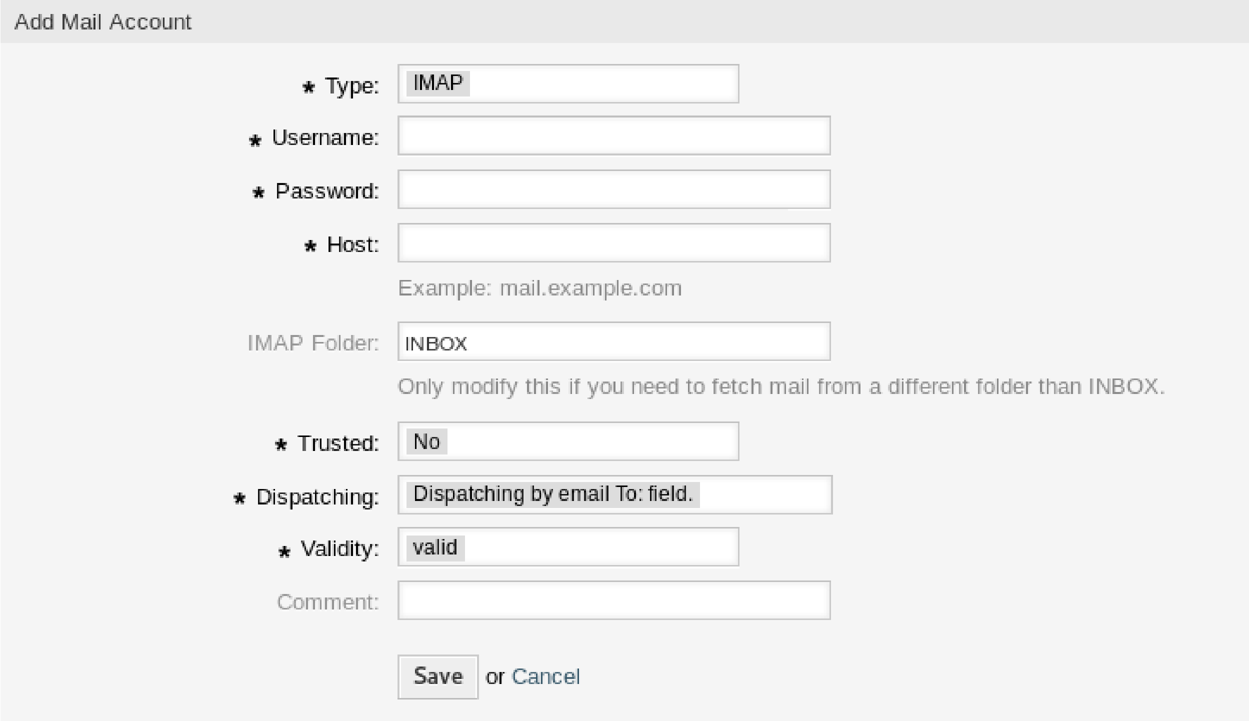
All data for the mail accounts are saved in the OTRS database. The `bin/otrs.Console.pl Maint::PostMaster::MailAccountFetch` command uses the settings in the database and fetches the mail. You can execute it manually to check if all your mail settings are working properly.

On a default installation, the mail is fetched every 10 minutes when the OTRS daemon is running.

3.7.1 Manage Mail Accounts

To add a mail account:

1. Click on the *Add Mail Account* button in the left sidebar.
2. Fill in the required fields.
3. Click on the *Save* button.



Add Mail Account

* Type:	IMAP
* Username:	
* Password:	
* Host:	 Example: mail.example.com
IMAP Folder:	INBOX Only modify this if you need to fetch mail from a different folder than INBOX.
* Trusted:	No
* Dispatching:	Dispatching by email To: field.
* Validity:	valid
Comment:	

Fig. 30: Add Mail Account Screen

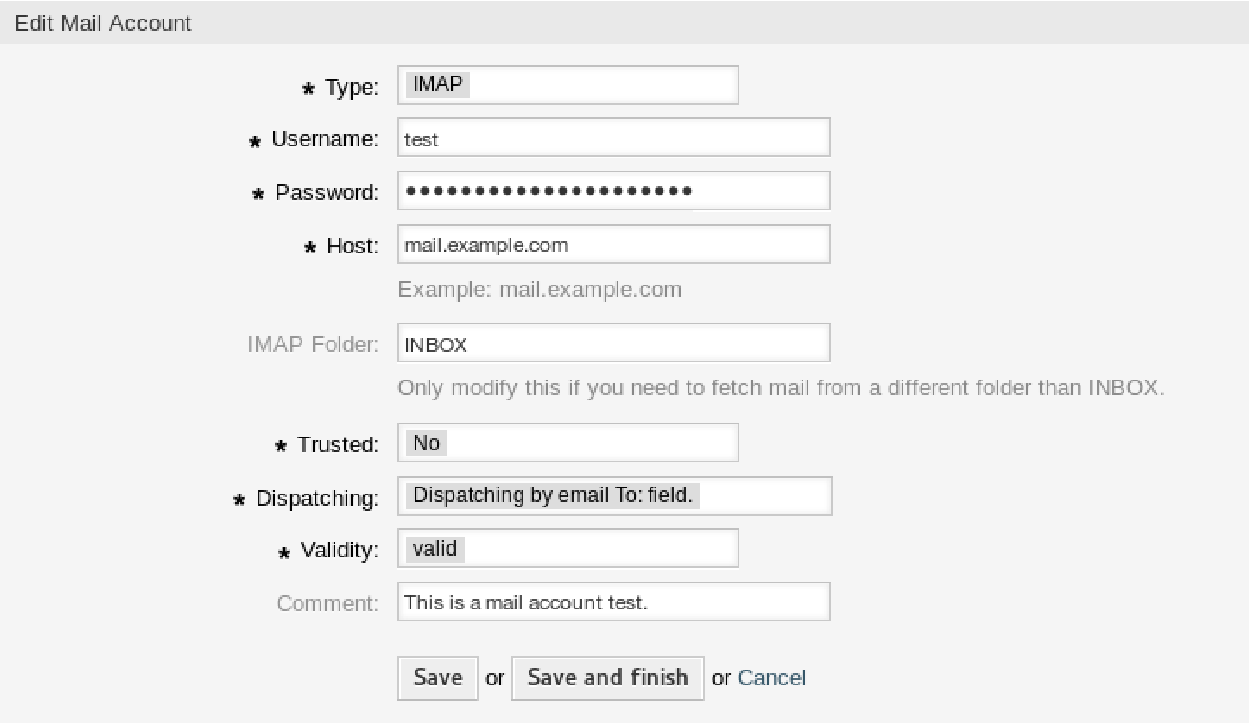
To edit a mail account:

1. Click on a mail account in the list of mail accounts.
2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.

To delete a mail account:

1. Click on the trash icon in the list of mail accounts.
2. Click on the *Confirm* button.

Note: If several mail accounts are added to the system, a filter box is useful to find a particular mail account by just typing to filter.



Edit Mail Account

* Type: IMAP

* Username: test

* Password: ••••••••••••••••••••

* Host: mail.example.com

Example: mail.example.com

IMAP Folder: INBOX

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

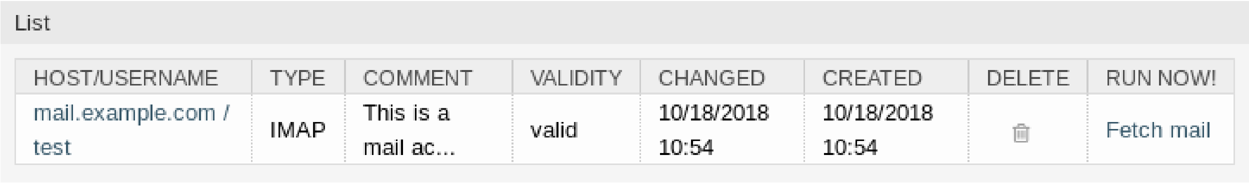
* Trusted: No

* Dispatching: Dispatching by email To: field.

* Validity: valid

Comment: This is a mail account test.

Fig. 31: Edit Mail Account Screen



List

HOST/USERNAME	TYPE	COMMENT	VALIDITY	CHANGED	CREATED	DELETE	RUN NOW!
mail.example.com / test	IMAP	This is a mail ac...	valid	10/18/2018 10:54	10/18/2018 10:54	🗑	Fetch mail

Fig. 32: Delete Mail Account Screen

3.7.2 Mail Account Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Type * There are different kind of protocols that are used for fetching mail. The following protocols are supported:

- IMAP
- IMAPS
- IMAPTLS
- POP3
- POP3S
- POP3TLS

Username * The username of the mail account.

Password * The password of the mail account.

> **Note:** Only regular password authentication is supported. For advanced OAuth2 support and in case you are an OTRS customer, please raise a support ticket in the OTRS customer portal.

Host * The host name of the mail account. Specify how the current article subject should be dealt with. The following methods are available:

IMAP Folder The folder in the mail account to be fetched. Other folders remain untouched.

Trusted If *Yes* is selected, any `X-OTRS` headers attached to an incoming message are evaluated and executed. Because the `X-OTRS` header can execute some actions in the ticket system, you should set this option to *Yes* only for known senders.

See also:

The `X-OTRS` headers are explained in the filter conditions of *Postmaster Filters*.

Dispatching The distribution of incoming messages can be controlled if they need to be sorted by queue or by the content of the *To:* field.

Dispatching by email To: field The system checks if a queue is linked with the address in the *To:* field of the incoming mail. You can link an address to a queue in the *Email Addresses* screen. If the address in the *To:* field is linked with a queue, the new message will be sorted into the linked queue. If no link is found between the address in the *To:* field and any queue, then the message flows into the *Raw* queue in the system, which is the postmaster default queue after a default installation.

See also:

The postmaster default queue can be changed in system configuration setting PostmasterDefaultQueue.

Dispatching by selected Queue All incoming messages will be sorted into the specified queue. The address where the mail was sent to is disregarded in this case.

Validity * Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

3.7.3 Setting up Incoming Emails

For incoming emails, OTRS can be configured in a variety of ways.The best option to do this might depend on your specific circumstances.

Receiving Mails via Local MDA

Using a local mail transfer agent (MTA) to receive mails directly is the **recommended** option. It provides near to real time mail delivery and the setup is more stable than when you poll your mails via IMAP(S) or POP(S). Authentication is also no longer an issue.

Redirecting mails to another address Your current mail server must relay the emails to the local MTA.

SMTP Transport An alternative router is setup on the receiving MTA to deliver mails from specific users to an external SMTP server.

In both cases, the use of *Procmail* is required as the mail delivery agent (MDA), as well as the proper `$HOME` setup for the user *otrs* on your system. An example file `.procmailrc.dist` can be used directly without modification. Just remove the `.dist` ending. Mails sent to the local *otrs* user are then piped to the console command for processing.

Warning: Please note that all "push" mail transferred directly to the `otrs` user (i.e. via the console command) is always trusted. This means `X-OTRS` headers in emails can and will affect the distribution of mails and setting of ticket properties. This includes the `X-OTRS-FollowUp` headers.

For more information, please read the *Postmaster Filters* chapter.

Note: Errors will send the mail to the directory `var/spool/problem-email`.

Read the Postfix documentation or the Exim4 documentation for more information on the setup.

Note: In case you are an OTRS customer, you can configure relaying by raising a support ticket in the OTRS customer portal.

Receiving Mails via External Email Provider

Alternatively, it is also possible to use an external email provider for incoming emails.

Note: Only regular password authentication is supported. For advanced OAuth2 support and in case you are an OTRS customer, please raise a support ticket in the OTRS customer portal.

To setup the system to use *Office 365* as email provider:

1. Read the POP and IMAP email settings for Outlook chapter in the official documentation.

2. Fill the form with the following data:
   - Type: select the desired protocol
   - Username: your email address
   - Password: your email password
   - Host: *outlook.office365.com*

Warning: On October 13th, 2020 Microsoft stops supporting and retires basic authentication for POP3 and IMAP. Only OAuth2 can be used for authentication. For more information, please read the original blog post.

To setup the system to use *Gmail* as email provider:

1. Read the Check Gmail through other email platforms chapter in the official documentation.
2. Fill the form with the following data:
   - Type: select the desired protocol
   - Username: your email address
   - Password: your email password
   - Host: *imap.gmail.com*

To setup the system to use another external email provider:

1. Read the official documentation of the external provider.
2. Fill the form with the following data:
   - Type: select the desired protocol
   - Username: your email address
   - Password: your email password
   - Host: the POP3 or IMAP address of the external provider

See also:

You can find detailed instructions in the *Email Addresses* chapter about how to configure outgoing emails.

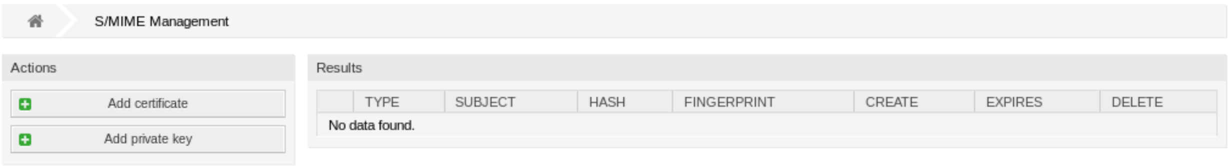
3.8 S/MIME Certificates

Faculty and staff have key roles safeguarding critical information by implementing information security policies, standards, and controls. Safe email communication is a vital part of protecting this communication.

OTRS empowers you to encrypt communications where needed by means of *S/MIME Certificates* or *PGP Keys*.

Note: Setup of services and software required for encryption are not covered here because of independence to this software.

Use this screen to add S/MIME certificates to the system. The S/MIME management screen is available in the *S/MIME Certificates* module of the *Communication & Notifications* group.

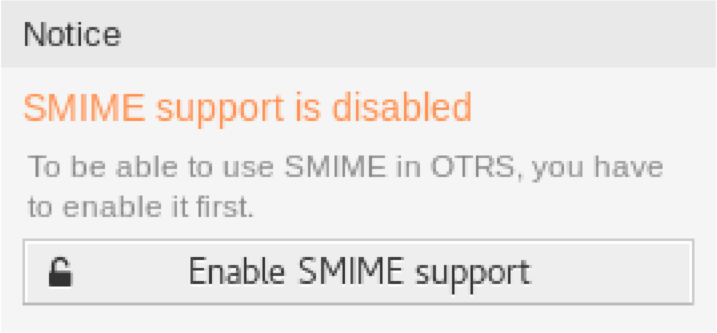


	TYPE	SUBJECT	HASH	FINGERPRINT	CREATE	EXPIRES	DELETE
No data found.							

Fig. 33: S/MIME Management Screen

3.8.1 Manage S/MIME Certificates

Note: To be able to use S/MIME certificates in OTRS, you have to activate its setting first.



Notice

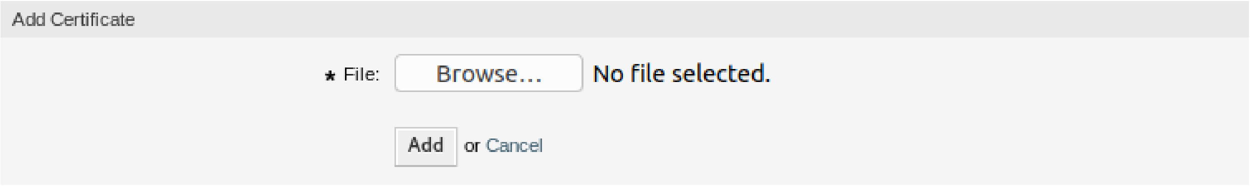
SMIME support is disabled

To be able to use SMIME in OTRS, you have to enable it first.

Fig. 34: Enable S/MIME Support

To add an S/MIME certificate:

1. Click on the *Add Certificate* button in the left sidebar.
2. Click on *Browse*… button to open the file dialog.
3. Select an S/MIME certificate from the file system.
4. Click on the *Add* button.



Add Certificate

* File: Browse… No file selected.

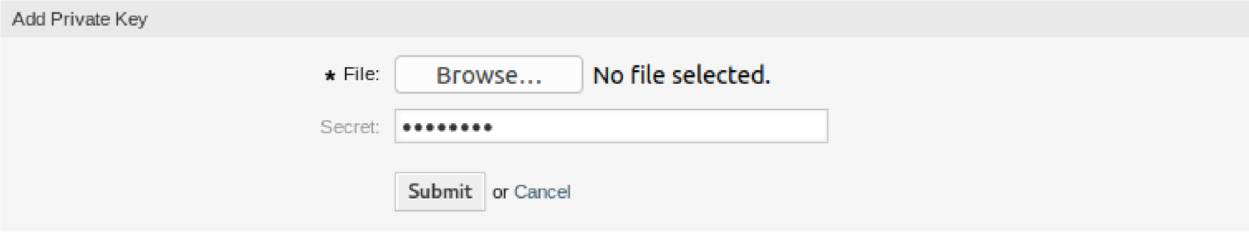
Fig. 35: Add S/MIME Certificate Screen

Note: Only non binary keys contain ASCII (Base64) armored data started with a `----- BEGIN CERTIFICATE -----` line can be uploaded which are most commonly `key.pem` or `root.crt`. Conversion of other formats like `cert.p7b` must be done using OpenSSL.

To add a private key:

1. Click on the *Add Private Key* button in the left sidebar.

2. Click on *Browse…* button to open the file dialog.
3. Select a private key from the file system.
4. Click on the *Submit* button.



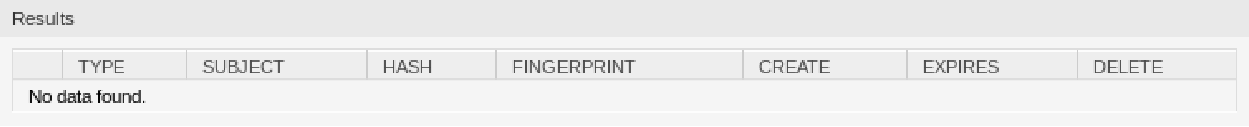
Add Private Key

⋆ File:	Browse… No file selected.
Secret:	••••••••
	Submit or Cancel

Fig. 36: Add S/MIME Private Key Screen

To delete an S/MIME certificate:

1. Click on the trash icon in the list of S/MIME certificates.
2. Click on the *Confirm* button.



Results

	TYPE	SUBJECT	HASH	FINGERPRINT	CREATE	EXPIRES	DELETE
No data found.							

Fig. 37: Delete S/MIME Certificate Screen

Note: If several S/MIME certificates are added to the system, use the filter box to find a particular S/MIME certificate by just typing the name to filter.

To adjust the S/MIME certificate settings of the system:

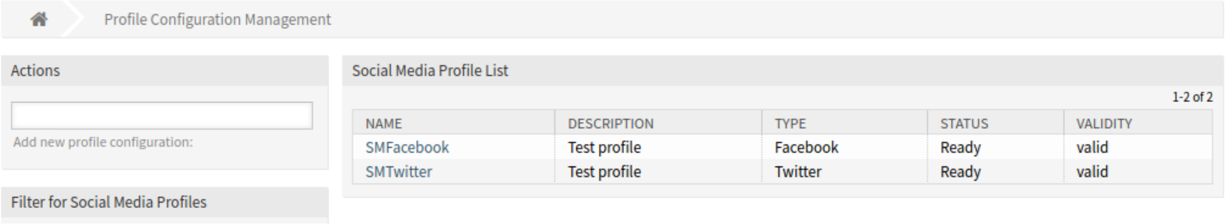
1. Go to *System Configuration* screen.
2. Navigate to *Core* → *Crypt* → *SMIME* in the navigation tree.
3. Review the settings.

3.9 Social Media Profiles

Note: This feature is available only for managed customers. A special contract is needed to use this feature.

Social media profile integration makes it possible to connect **OTRS** with social media channels. This allows agents to send and receive messages and communicate with social media users like with customer users. This feature stores the messages in the tickets as articles.

Use this screen to add social media profiles to the system. The social media profile management screen is available in the *Social Media Profiles* module of the *Communication & Notifications* group.



Profile Configuration Management
Actions
Add new profile configuration:
Filter for Social Media Profiles
Social Media Profile List
1-2 of 2

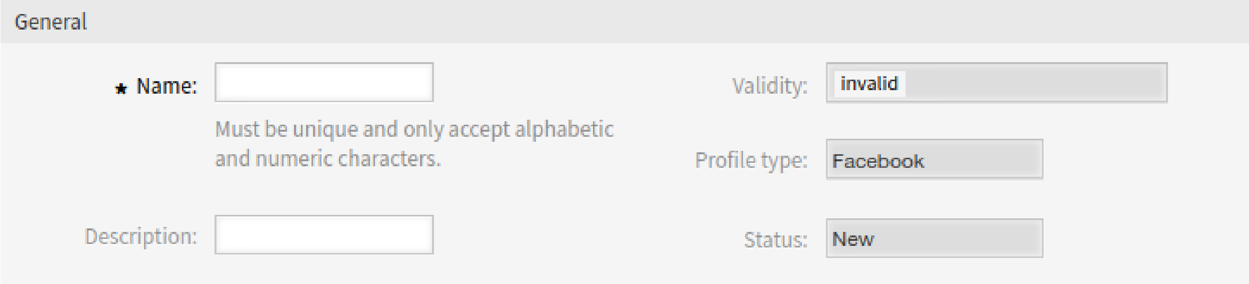
NAME	DESCRIPTION	TYPE	STATUS	VALIDITY
SMFacebook	Test profile	Facebook	Ready	valid
SMTwitter	Test profile	Twitter	Ready	valid

Fig. 38: Profile Configuration Management Screen

3.9.1 Manage Facebook Profiles

To add a Facebook profile:

1. Select *Facebook* from the *Actions* widget in the left sidebar.
2. Fill in the required fields in the *General* widget.



* Name:

Must be unique and only accept alphabetic and numeric characters.

Description:

Validity: invalid

Profile type: Facebook

Status: New

Fig. 39: General Widget

Name * The name of this resource. Must be unique and only accept alphabetic and numeric characters. The name will be displayed in the overview table.

Description Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

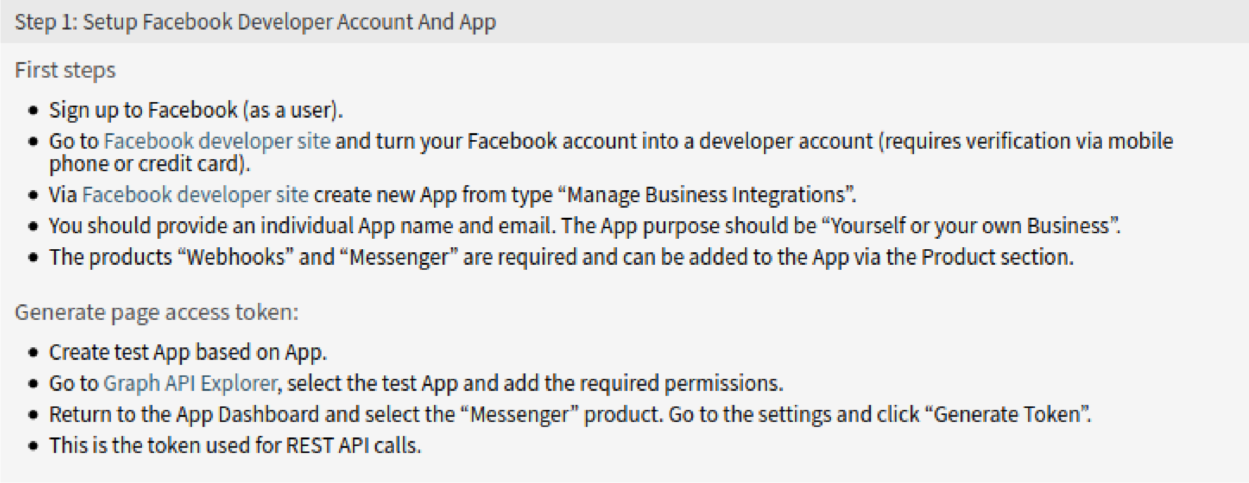
3. Click on the *Save* button and follow the instruction steps. You have to save the entered data between all steps.
4. Read the instructions and create the necessary Facebook resources.
5. Add the access keys.

   App ID * Paste the “App ID”. This is located in basic settings of the Facebook App.

   App Secret * Reveal and paste the “App Secret”. This is located in basic settings of the Facebook App.

   Access Token * Paste the generated token from your Facebook App.

6. Click on the *Check Access* button to test the configuration.
7. Click on the *Check Permissions* button to test the permissions.
8. Setup Facebook webhooks for the app.
9. Select the *Use production mode* checkbox if the Facebook app is ready for use in production.
10. Click on the *Verify webhook setup* button to test the configuration.



Step 1: Setup Facebook Developer Account And App

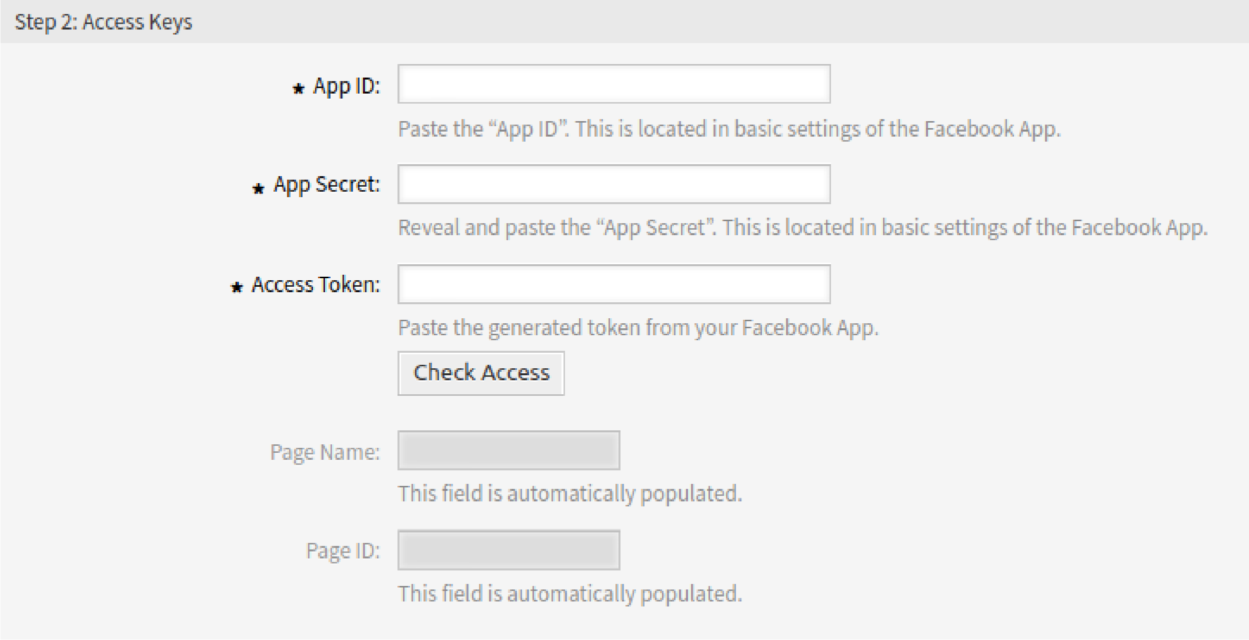
First steps

- Sign up to Facebook (as a user).
- Go to Facebook developer site and turn your Facebook account into a developer account (requires verification via mobile phone or credit card).
- Via Facebook developer site create new App from type "Manage Business Integrations".
- You should provide an individual App name and email. The App purpose should be "Yourself or your own Business".
- The products "Webhooks" and "Messenger" are required and can be added to the App via the Product section.

Generate page access token:

- Create test App based on App.
- Go to Graph API Explorer, select the test App and add the required permissions.
- Return to the App Dashboard and select the "Messenger" product. Go to the settings and click "Generate Token".
- This is the token used for REST API calls.

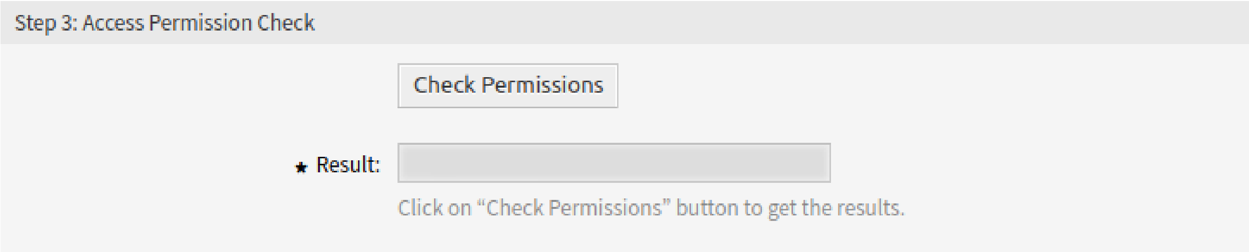
Fig. 40: Step 1: Setup Facebook Developer Account And App



Step 2: Access Keys

★ App ID:	Paste the "App ID". This is located in basic settings of the Facebook App.
★ App Secret:	Reveal and paste the "App Secret". This is located in basic settings of the Facebook App.
★ Access Token:	Paste the generated token from your Facebook App.
	Check Access
Page Name:	This field is automatically populated.
Page ID:	This field is automatically populated.

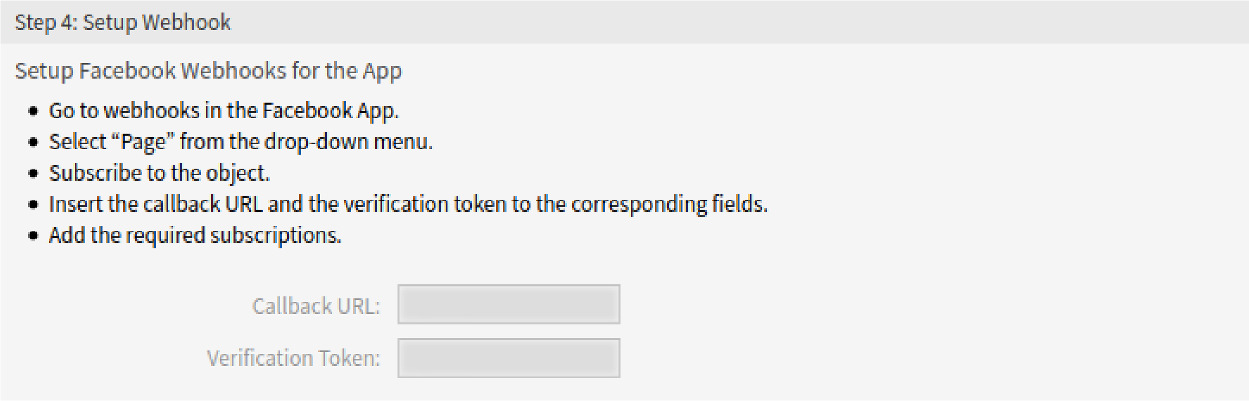
Fig. 41: Step 2: Access Keys



Step 3: Access Permission Check

	Check Permissions
★ Result:	Click on "Check Permissions" button to get the results.

Fig. 42: Step 3: Access Permission Check



Step 4: Setup Webhook

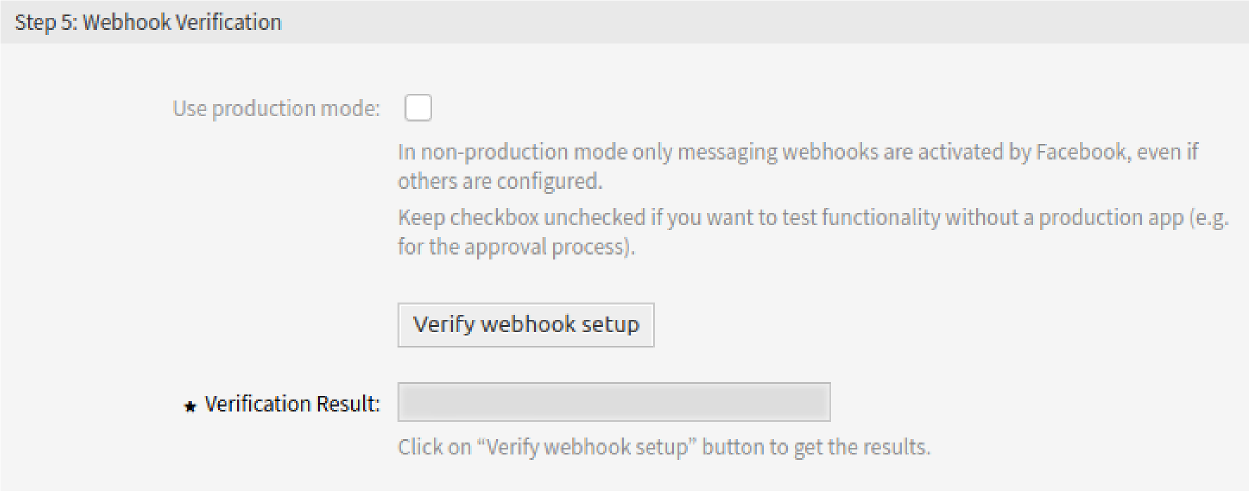
Setup Facebook Webhooks for the App

- Go to webhooks in the Facebook App.
- Select “Page” from the drop-down menu.
- Subscribe to the object.
- Insert the callback URL and the verification token to the corresponding fields.
- Add the required subscriptions.

Callback URL:

Verification Token:

Fig. 43: Step 4: Setup Webhook



Step 5: Webhook Verification

Use production mode: □

In non-production mode only messaging webhooks are activated by Facebook, even if others are configured.

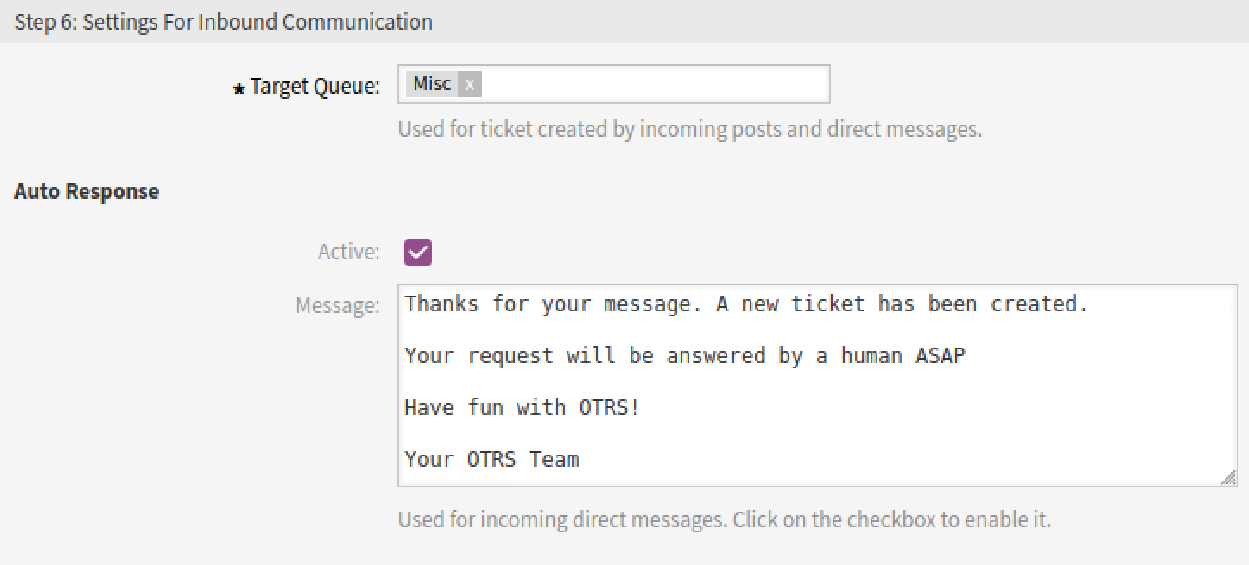
Keep checkbox unchecked if you want to test functionality without a production app (e.g. for the approval process).

* Verification Result:

Click on “Verify webhook setup” button to get the results.

Fig. 44: Step 5: Webhook Verification

11. Select a target queue and add automatic response for the messages from Facebook.



Step 6: Settings For Inbound Communication

★ Target Queue: Misc x

Used for ticket created by incoming posts and direct messages.

Auto Response

Active: ☑

Message:

```
Thanks for your message. A new ticket has been created.

Your request will be answered by a human ASAP

Have fun with OTRS!

Your OTRS Team
```

Used for incoming direct messages. Click on the checkbox to enable it.

Fig. 45: Step 6: Settings For Inbound Communication

Target Queue * Select the queue used for incoming entities.

Auto Response Add an automatic response used for incoming direct messages. Click on the checkbox to enable it.

12. Click on the *Save* or *Save and Finish* button.

> **Warning:** Social media profiles can not be deleted from the system. They can only be deactivated by setting the *Validity* option to *invalid* or *invalid-temporarily*.

To edit a Facebook profile:

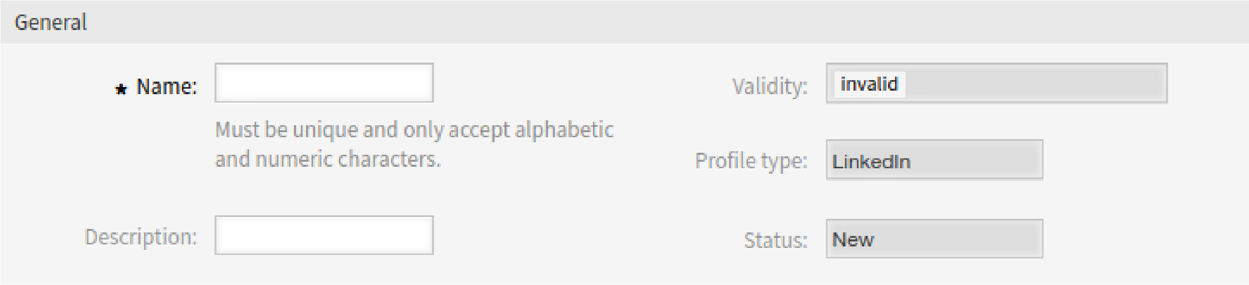
1. Click on a Facebook profile in the list of social media profiles.
2. Modify the fields. Use the *Previous Step* and the *Next Step* buttons to navigate between the steps.
3. Click on the *Save* or *Save and finish* button.

> **Note:** If several social media profiles are added to the system, use the filter box to find a particular social media profile by just typing the name to filter.

3.9.2 Manage LinkedIn Profiles

To add a LinkedIn profile:

1. Select *LinkedIn* from the *Actions* widget in the left sidebar.
2. Fill in the required fields in the *General* widget.



General

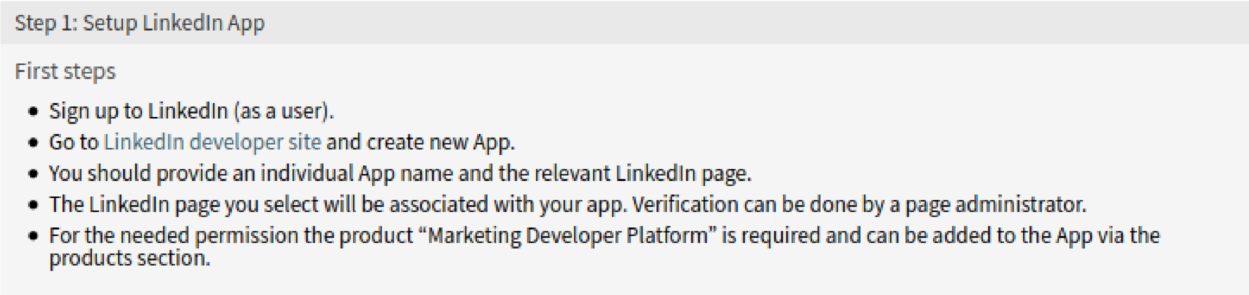
★ Name:		Validity:	invalid
	Must be unique and only accept alphabetic and numeric characters.	Profile type:	LinkedIn
Description:		Status:	New

Fig. 46: General Widget

Name * The name of this resource. Must be unique and only accept alphabetic and numeric characters. The name will be displayed in the overview table.

Description Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

3. Click on the *Save* button and follow the instruction steps. You have to save the entered data between all steps.
4. Read the instructions and create the necessary LinkedIn resources.



Step 1: Setup LinkedIn App

First steps

- Sign up to LinkedIn (as a user).
- Go to LinkedIn developer site and create new App.
- You should provide an individual App name and the relevant LinkedIn page.
- The LinkedIn page you select will be associated with your app. Verification can be done by a page administrator.
- For the needed permission the product "Marketing Developer Platform" is required and can be added to the App via the products section.

Fig. 47: Step 1: Setup LinkedIn App

5. Configure the application.

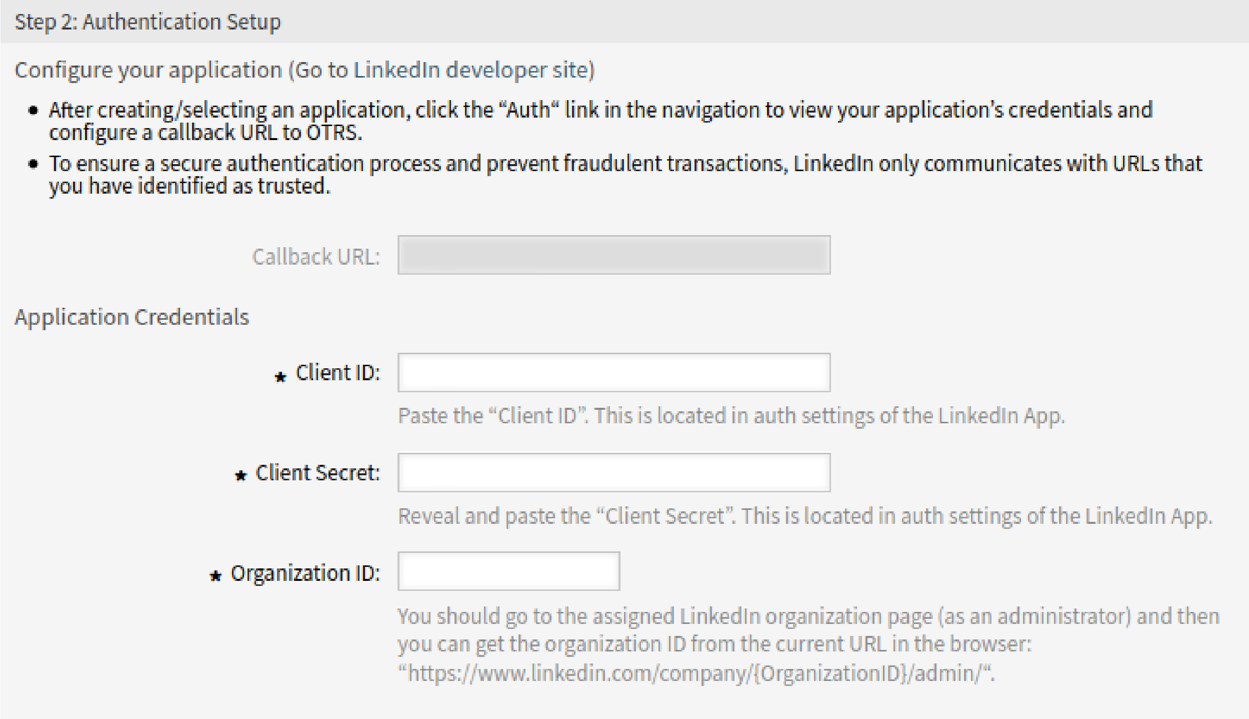
   Client ID * Paste the "Client ID". This is located in auth settings of the LinkedIn App.

   Client Secret * Reveal and paste the "Client Secret". This is located in auth settings of the LinkedIn App.

   Organization ID * You should go to the assigned LinkedIn organization page (as an administrator) and then you can get the organization ID from the current URL in the browser: "https://www.linkedin.com/company/{OrganizationID}/admin/ ".

6. Click on the *Save* button to test the configuration.
7. Click on the *Approve Application* button to test the permissions.

   Once redirected, you will be presented with a LinkedIn's authentication screen. This identifies your application and outlines the particular member permissions that your application is requesting. By providing valid LinkedIn credentials and clicking "Allow ", you approves the application's request to access the member data and interact with LinkedIn on their behalf. This approval instructs LinkedIn to redirect you to the callback URL that you defined before.



Step 2: Authentication Setup

Configure your application (Go to LinkedIn developer site)

- After creating/selecting an application, click the "Auth" link in the navigation to view your application's credentials and configure a callback URL to OTRS.
- To ensure a secure authentication process and prevent fraudulent transactions, LinkedIn only communicates with URLs that you have identified as trusted.

Callback URL:

Application Credentials

★ Client ID:

Paste the "Client ID". This is located in auth settings of the LinkedIn App.

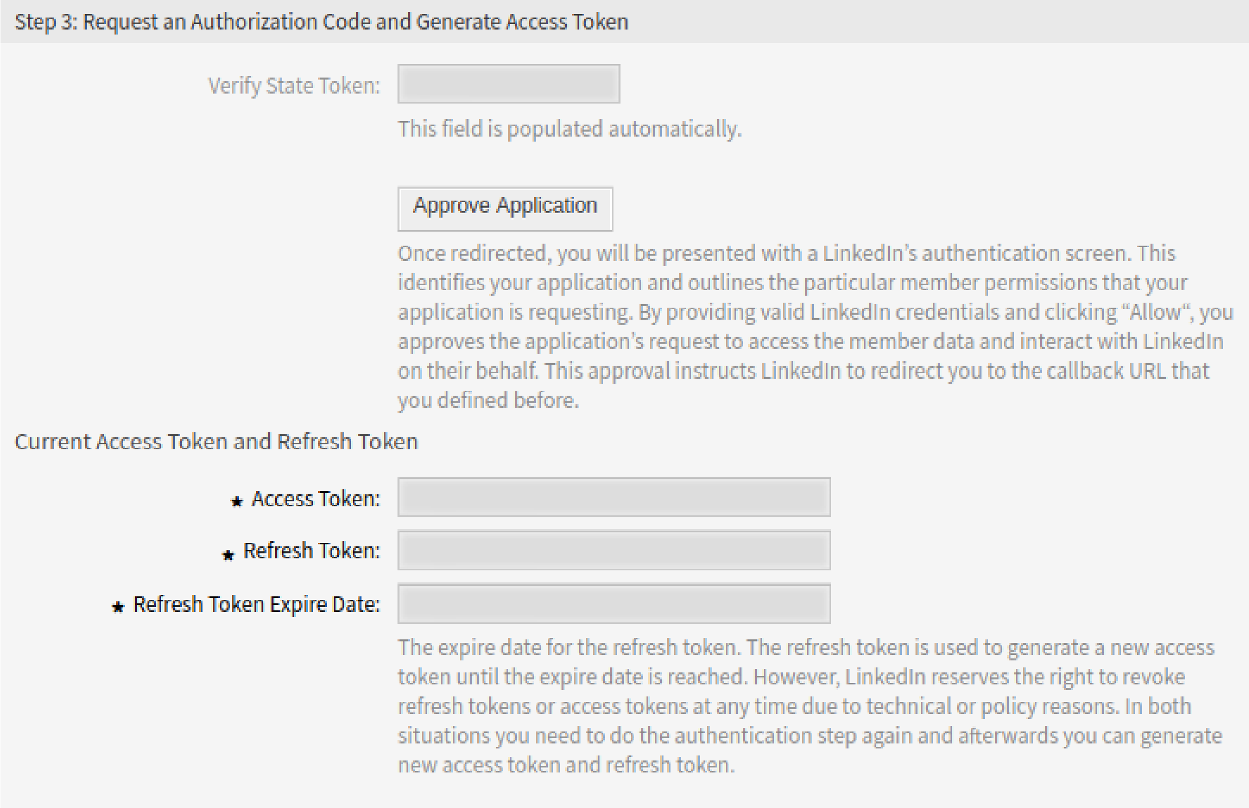
★ Client Secret:

Reveal and paste the "Client Secret". This is located in auth settings of the LinkedIn App.

★ Organization ID:

You should go to the assigned LinkedIn organization page (as an administrator) and then you can get the organization ID from the current URL in the browser: "https://www.linkedin.com/company/{OrganizationID}/admin/".

Fig. 48: Step 2: Authentication Setup



Step 3: Request an Authorization Code and Generate Access Token

Verify State Token:

This field is populated automatically.

Once redirected, you will be presented with a LinkedIn's authentication screen. This identifies your application and outlines the particular member permissions that your application is requesting. By providing valid LinkedIn credentials and clicking "Allow", you approves the application's request to access the member data and interact with LinkedIn on their behalf. This approval instructs LinkedIn to redirect you to the callback URL that you defined before.

Current Access Token and Refresh Token

★ Access Token:

★ Refresh Token:

★ Refresh Token Expire Date:

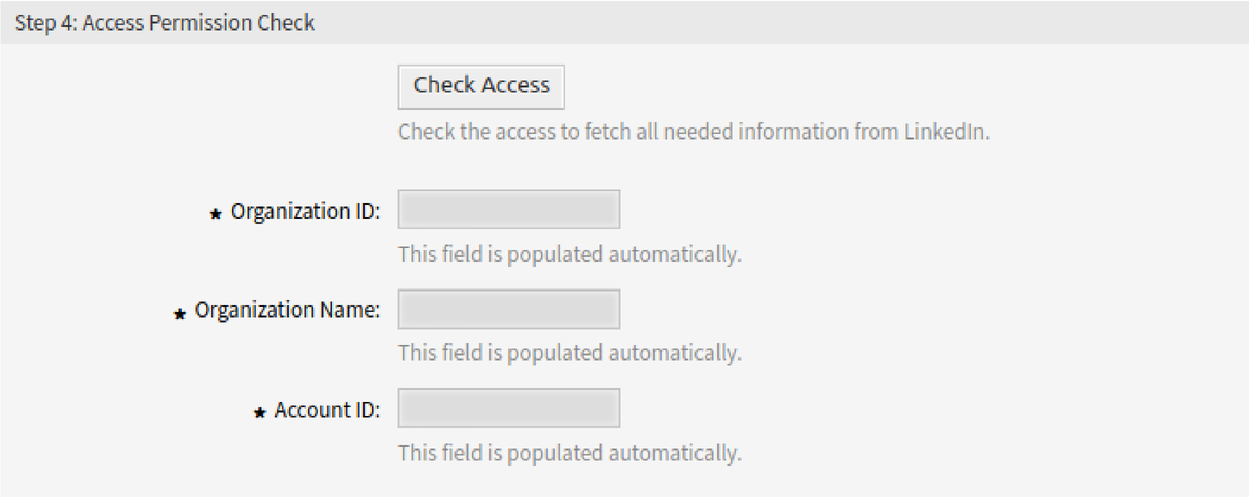
The expire date for the refresh token. The refresh token is used to generate a new access token until the expire date is reached. However, LinkedIn reserves the right to revoke refresh tokens or access tokens at any time due to technical or policy reasons. In both situations you need to do the authentication step again and afterwards you can generate new access token and refresh token.

Fig. 49: Step 3: Request an Authorization Code and generate Access Token

The expire date for the refresh token. The refresh token is used to generate a new access token until the expire date is reached. However, LinkedIn reserves the right to revoke Refresh Tokens or Access Tokens at any time due to technical or policy reasons. In both situations you need to do the authentication step again and afterwards you can generate a new access token and refresh token.

Note: For security reasons, the authorization code expires in one year. A notification will be displayed in the administrator interface 1 month before the refresh token will expire with a link to the profile management.

8. Click on the *Check Access* button to fetch all needed information from LinkedIn.



Step 4: Access Permission Check

Check the access to fetch all needed information from LinkedIn.

★ Organization ID:

This field is populated automatically.

★ Organization Name:

This field is populated automatically.

★ Account ID:

This field is populated automatically.

Fig. 50: Step 4: Access Permission Check

9. Click on the *Save* or *Save and Finish* button.

Warning: Social media profiles can not be deleted from the system. They can only be deactivated by setting the *Validity* option to *invalid* or *invalid-temporarily*.

To edit a LinkedIn profile:

1. Click on a LinkedIn profile in the list of social media profiles.
2. Modify the fields. Use the *Previous Step* and the *Next Step* buttons to navigate between the steps.
3. Click on the *Save* or *Save and finish* button.

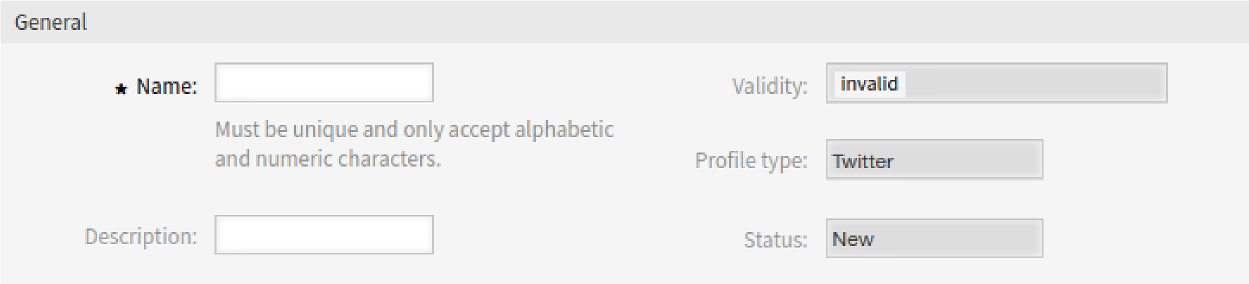
Note: If several social media profiles are added to the system, use the filter box to find a particular social media profile by just typing the name to filter.

3.9.3 Manage Twitter Profiles

To add a Twitter profile:

1. Select *Twitter* from the *Actions* widget in the left sidebar.

2. Fill in the required fields in the *General* widget.



General

★ Name:

Must be unique and only accept alphabetic and numeric characters.

Description:

Validity: invalid

Profile type: Twitter

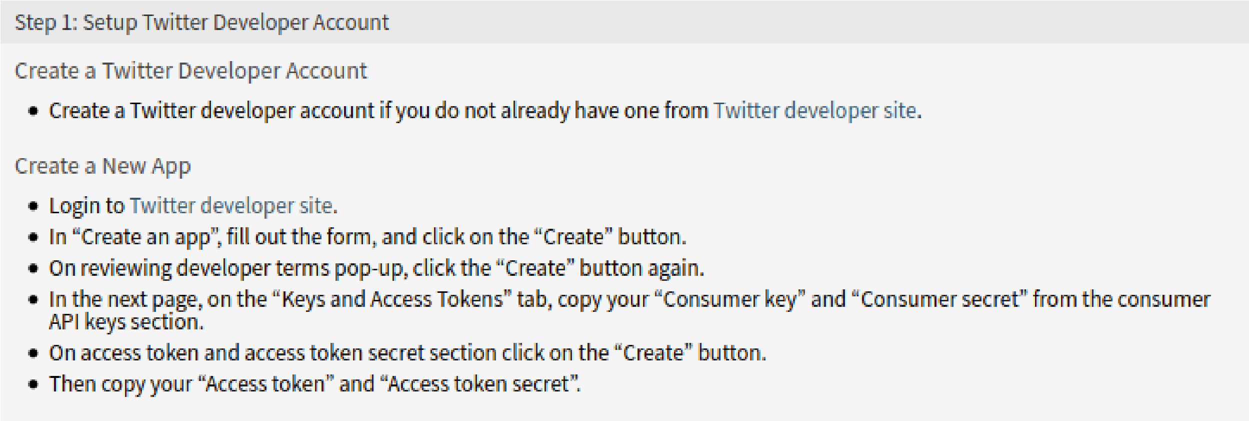
Status: New

Fig. 51: General Widget

Name * The name of this resource. Must be unique and only accept alphabetic and numeric characters. The name will be displayed in the overview table.

Description Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

3. Click on the *Save* button and follow the instruction steps. You have to save the entered data between all steps.
4. Read the instructions and create the necessary Twitter resources.



Step 1: Setup Twitter Developer Account

Create a Twitter Developer Account

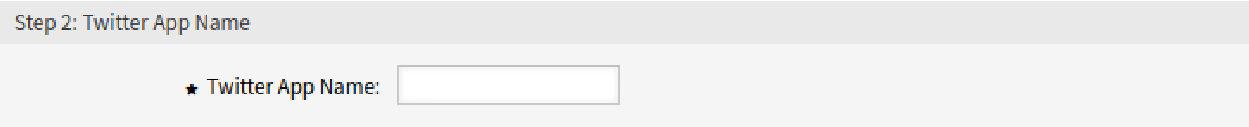
- Create a Twitter developer account if you do not already have one from Twitter developer site.

Create a New App

- Login to Twitter developer site.
- In "Create an app", fill out the form, and click on the "Create" button.
- On reviewing developer terms pop-up, click the "Create" button again.
- In the next page, on the "Keys and Access Tokens" tab, copy your "Consumer key" and "Consumer secret" from the consumer API keys section.
- On access token and access token secret section click on the "Create" button.
- Then copy your "Access token" and "Access token secret".

Fig. 52: Step 1: Setup Twitter Developer Account

5. Add the Twitter app name.



Step 2: Twitter App Name

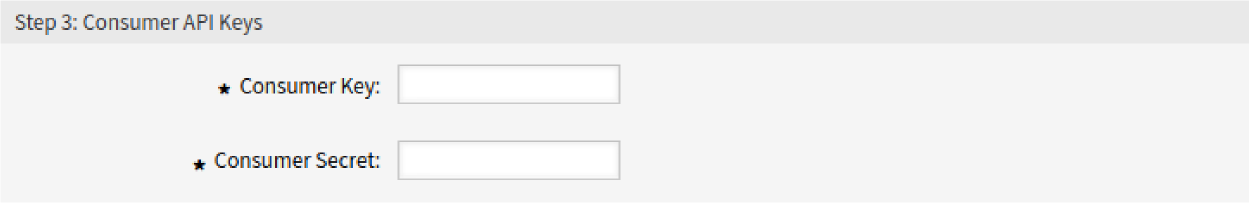
★ Twitter App Name:

Fig. 53: Step 2: Twitter App Name

Twitter App Name * Enter a Twitter app name.

6. Add the consumer API keys.

Consumer Key * Copy and paste your consumer key from the consumer API key section of the Twitter app.


Step 3: Consumer API Keys

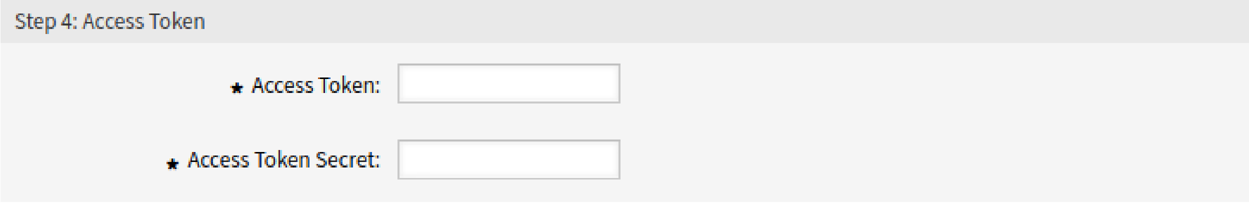
★ Consumer Key:

★ Consumer Secret:

Fig. 54: Step 3: Access Permissions

Consumer Secret * Copy and paste your consumer secret from the consumer API key section of the Twitter app.

7. Add the access token.


Step 4: Access Token

★ Access Token:

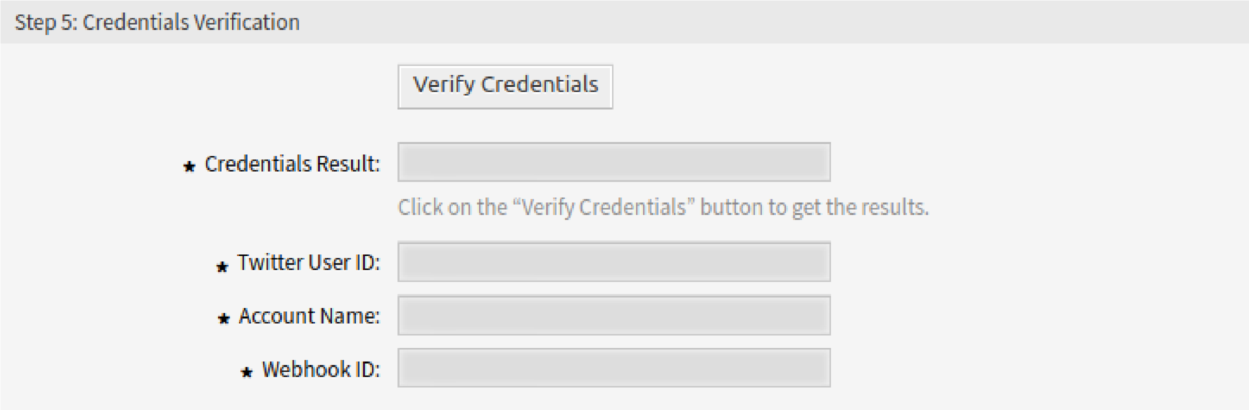
★ Access Token Secret:

Fig. 55: Step 4: Access Token

Access Token * Copy and paste your access token from the Twitter app.

Access Token Secret * Copy and paste your access token secret from the Twitter app.

8. Click on the *Verify Credentials* button to test the configuration.


Step 5: Credentials Verification

★ Credentials Result:

Click on the "Verify Credentials" button to get the results.

★ Twitter User ID:

★ Account Name:

★ Webhook ID:

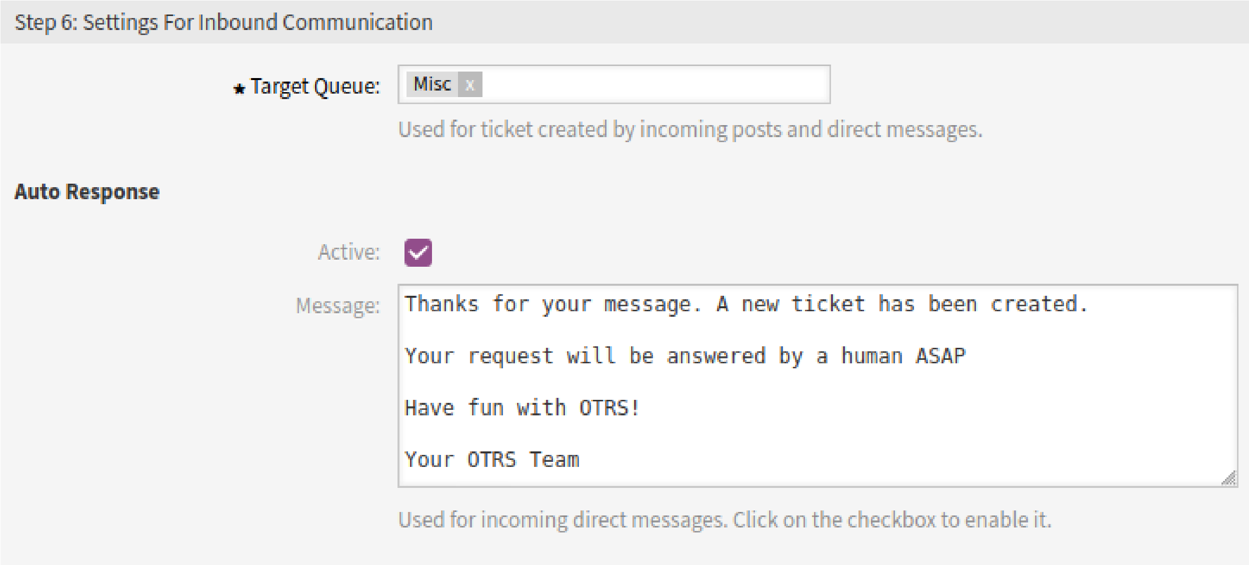
Fig. 56: Step 5: Credentials Verification

10. Select a target queue and add automatic response for the messages from Twitter.

Target Queue * Select the queue used for incoming entities.

Auto Response Add an automatic response used for incoming direct messages. Click on the checkbox to enable it.

12. Click on the *Save and Finish* button.



Step 6: Settings For Inbound Communication

★ Target Queue: Misc

Used for ticket created by incoming posts and direct messages.

Auto Response

Active: ☑

Message:

```
Thanks for your message. A new ticket has been created.

Your request will be answered by a human ASAP

Have fun with OTRS!

Your OTRS Team
```

Used for incoming direct messages. Click on the checkbox to enable it.

Fig. 57: Step 6: Incoming Settings

Warning: Social media profiles can not be deleted from the system. They can only be deactivated by setting the *Validity* option to *invalid* or *invalid-temporarily*.

To edit a Twitter profile:

1. Click on a Twitter profile in the list of social media profiles.
2. Modify the fields. Use the *Previous Step* and the *Next Step* buttons to navigate between the steps.
3. Click on the *Save* or *Save and finish* button.

Note: If several social media profiles are added to the system, use the filter box to find a particular social media profile by just typing the name to filter.

3.9.4 Limit User Access to Profiles

In case you would like to limit the user access to a certain social media profile, you can use the *Access Control Lists (ACL)*.

In the following example, we restrict the access of members of the group `facebook` to a Facebook page profile with relevant views. This guide assumes that you have already created this group and assigned users with *rw* permissions to it.

To identify the profile ID:

1. Go to the *Social Media Profiles* module in the administrator interface.
2. Select the target profile in the list you would like to limit the access to.
3. Identify the `ID` of the current profile in the URL of the current page. In our case, the `ID` is *1*.

```
...;ProfileType=Facebook;ID=1
```

To create an ACL:

1. Go to the *Access Control Lists (ACL)* module in the administrator interface.
2. Click on the *Create New ACL* button in the left sidebar.
3. Fill in the required *Name* field with a suitable text.
4. Set *Validity* to *valid*.
5. Click on the *Save* button.

To edit and deploy the ACL:

1. Click in the *Match settings* field in the *Edit ACL structure* widget, and select `Properties`.
2. Click in the field below *Properties* and select `User`.
3. Type in `Group_rw` in the empty box below *User*.
4. Select `Negative Exact Match` from the drop-down list below *Group_rw* and enter the name of the group in the empty field next to it. In this example this is `facebook`.
5. Click in the field *Change settings* and select `PossibleNot`.
6. Click in the field below *PossibleNot* and select `Ticket`.
7. Type in `FacebookProfileID` in the empty box below *Ticket*.
8. In the empty box below *FacebookProfileID* type in the identified value of the profile. In our case, this is *1*.
9. Click on the *Save and finish* button.
10. Click on the *Deploy ACLs* button in the left sidebar.

From now on, a user who is not member of the group `facebook` will no longer have access to the page profile with the ID *1* in the *Facebook Page* field on the corresponding screens.

In case of a Twitter account, use a different group name (e.g. `twitter`) and use `TwitterProfileID` for the field name in step 7 of the ACL creation.

3.10 Ticket Notifications

Streamlining communication can save hours of labor and prevent mistakes. Sending certain messages at pre-defined stages of communication not only keeps the customer and agents informed about specific events, but it can also aid your agents by programmatically doing automated updates to the customer.

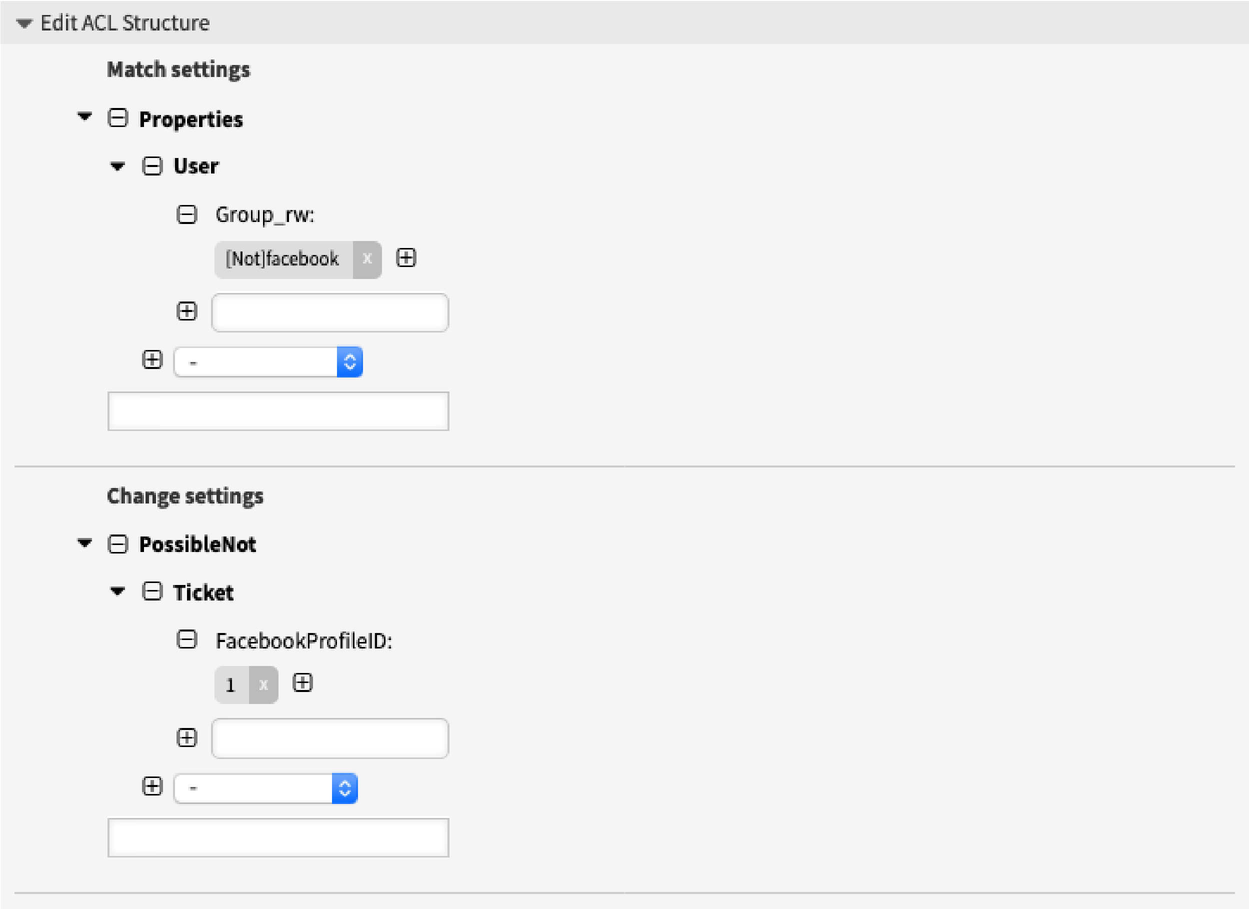
The flexible OTRS is an industry leader in email communication and offers you complete control of notifications based on any event in your system.

Use this screen to add ticket notifications to the system. In a fresh OTRS installation several ticket notifications are already added by default. The ticket notification management screen is available in the *Ticket Notifications* module of the *Communication & Notifications* group.

3.10.1 Manage Ticket Notifications

To add a ticket notification:

1. Click on the *Add Notification* button in the left sidebar.
2. Fill in the required fields as explained in *Ticket Notification Settings*.



Edit ACL Structure

Match settings

Properties

User

Group_rw:

[Not]facebook

Change settings

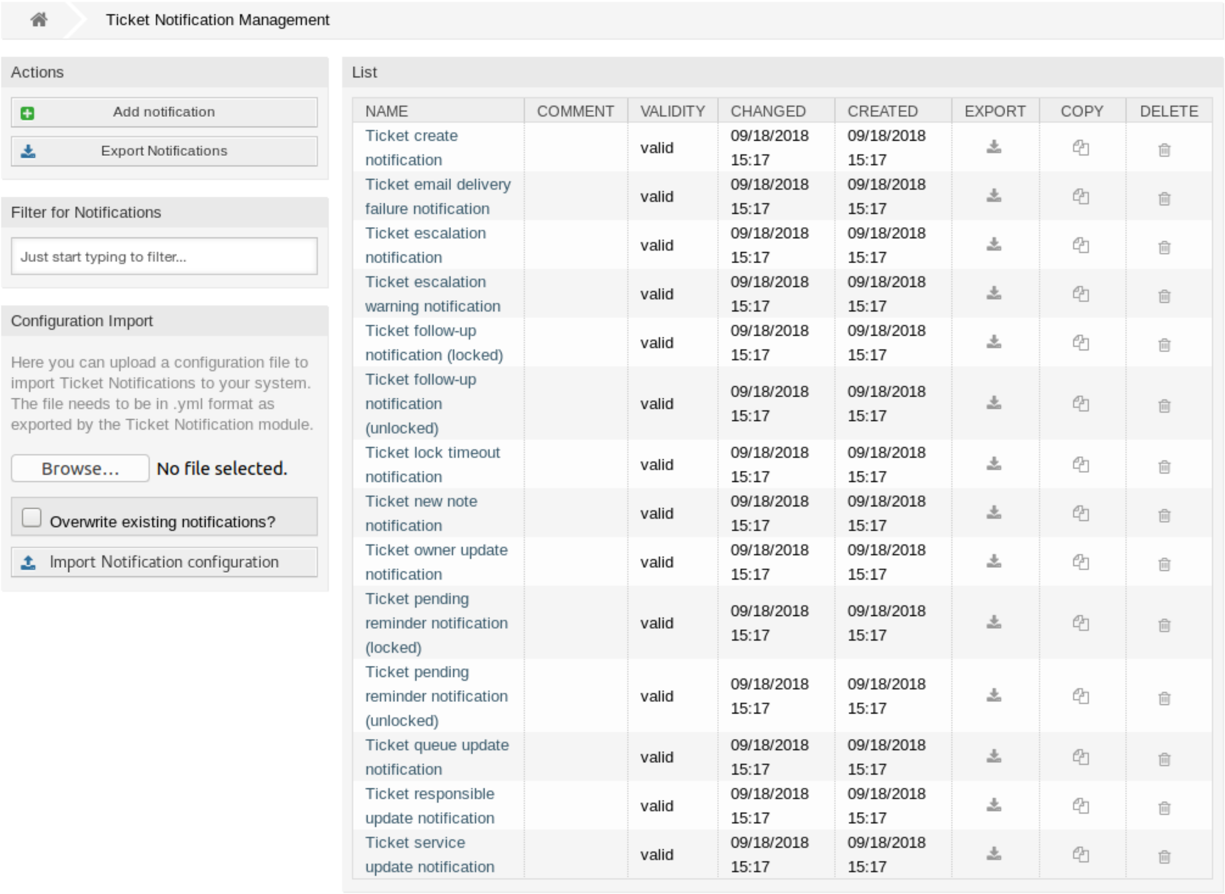
PossibleNot

Ticket

FacebookProfileID:

1

Fig. 58: ACL Rule to Limit Access to Facebook Profile



Ticket Notification Management
Actions
Add notification
Export Notifications
Filter for Notifications
Just start typing to filter...
Configuration Import

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

Browse... No file selected.
□ Overwrite existing notifications?
Import Notification configuration
List

NAME	COMMENT	VALIDITY	CHANGED	CREATED	EXPORT	COPY	DELETE
Ticket create notification		valid	09/18/2018 15:17	09/18/2018 15:17	⤓	⧉	🗑
Ticket email delivery failure notification		valid	09/18/2018 15:17	09/18/2018 15:17	⤓	⧉	🗑
Ticket escalation notification		valid	09/18/2018 15:17	09/18/2018 15:17	⤓	⧉	🗑
Ticket escalation warning notification		valid	09/18/2018 15:17	09/18/2018 15:17	⤓	⧉	🗑
Ticket follow-up notification (locked)		valid	09/18/2018 15:17	09/18/2018 15:17	⤓	⧉	🗑
Ticket follow-up notification (unlocked)		valid	09/18/2018 15:17	09/18/2018 15:17	⤓	⧉	🗑
Ticket lock timeout notification		valid	09/18/2018 15:17	09/18/2018 15:17	⤓	⧉	🗑
Ticket new note notification		valid	09/18/2018 15:17	09/18/2018 15:17	⤓	⧉	🗑
Ticket owner update notification		valid	09/18/2018 15:17	09/18/2018 15:17	⤓	⧉	🗑
Ticket pending reminder notification (locked)		valid	09/18/2018 15:17	09/18/2018 15:17	⤓	⧉	🗑
Ticket pending reminder notification (unlocked)		valid	09/18/2018 15:17	09/18/2018 15:17	⤓	⧉	🗑
Ticket queue update notification		valid	09/18/2018 15:17	09/18/2018 15:17	⤓	⧉	🗑
Ticket responsible update notification		valid	09/18/2018 15:17	09/18/2018 15:17	⤓	⧉	🗑
Ticket service update notification		valid	09/18/2018 15:17	09/18/2018 15:17	⤓	⧉	🗑

Fig. 59: Ticket Notification Management Screen

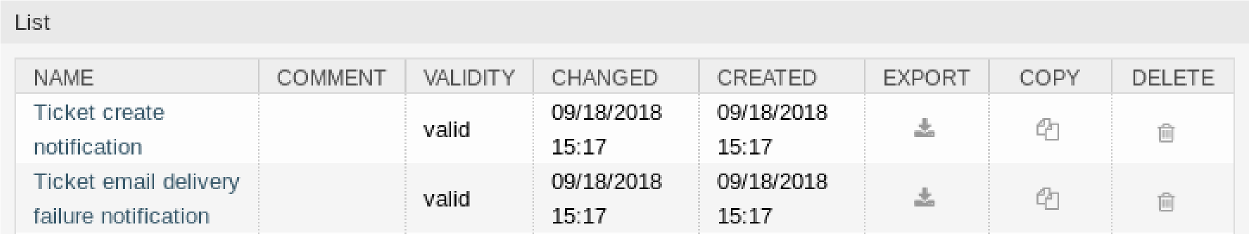
3. Click on the *Save* button.

To edit a ticket notification:

1. Click on a ticket notification in the list of ticket notifications.
2. Modify the fields as explained in *Ticket Notification Settings*.
3. Click on the *Save* or *Save and finish* button.

To delete a ticket notification:

1. Click on the trash icon in the list of ticket notifications.
2. Click on the *Confirm* button.



List

NAME	COMMENT	VALIDITY	CHANGED	CREATED	EXPORT	COPY	DELETE
Ticket create notification		valid	09/18/2018 15:17	09/18/2018 15:17	⭳	⧉	🗑
Ticket email delivery failure notification		valid	09/18/2018 15:17	09/18/2018 15:17	⭳	⧉	🗑

Fig. 60: Delete Ticket Notification Screen

To export all ticket notifications:

1. Click on the *Export Notifications* button in the left sidebar.
2. Choose a location in your computer to save the `Export_Notification.yml` file.

> **Warning:** Certain settings are exported as numeric IDs and will break when importing to a system where these settings do not appear or reference other named items.

To import ticket notifications:

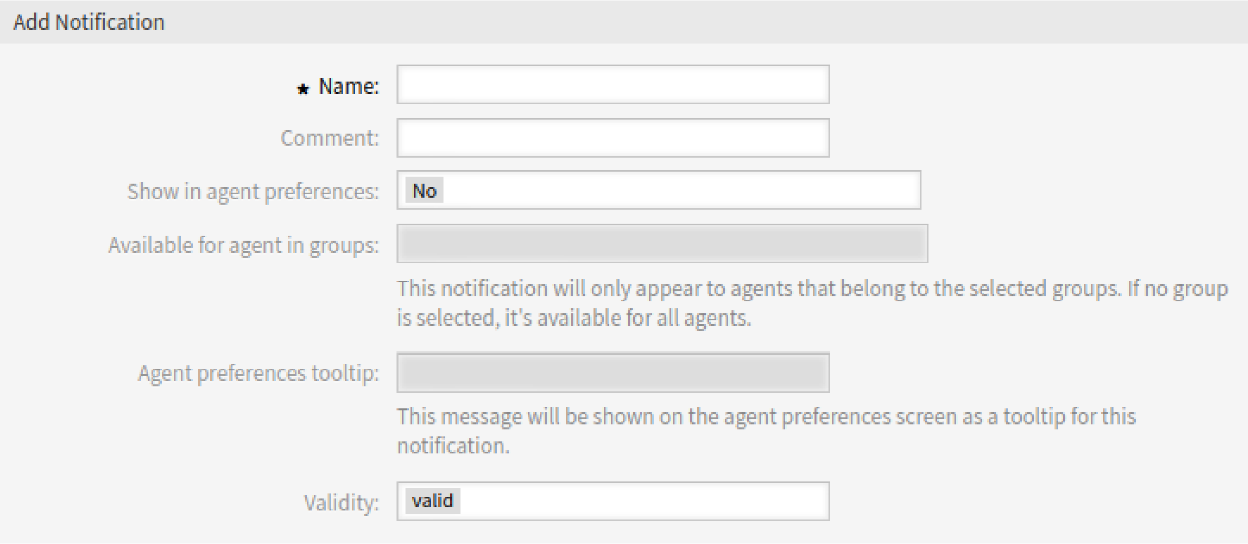
1. Click on the *Browse…* button in the left sidebar.
2. Select a previously exported `.yml` file.
3. Click on the *Overwrite existing notifications?* checkbox, if you would like to overwrite the existing notifications.
4. Click on the *Import Notification configuration* button.

3.10.2 Ticket Notification Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

See also:

For an example, see a default ticket notification which is included in a fresh OTRS installation.



Add Notification

★ Name:

Comment:

Show in agent preferences: No

Available for agent in groups:

This notification will only appear to agents that belong to the selected groups. If no group is selected, it's available for all agents.

Agent preferences tooltip:

This message will be shown on the agent preferences screen as a tooltip for this notification.

Validity: valid

Fig. 61: Ticket Notification Settings - Basic

Basic Ticket Notification Settings

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

Show in agent preferences Define how the notification should be displayed in agent preferences. The following options are available:

> **No** The notification won’ t be displayed in agent preferences. The notification is sent to all appropriate agents by the defined method.
>
> **Yes** The notification will be displayed in agent preferences for selection. The agents may opt-in or opt-out.
>
> **Yes, but require at least one active notification method.** The notification will be displayed in agent preferences, but require at least one active notification method. This is annotated by an asterisk next to the name.

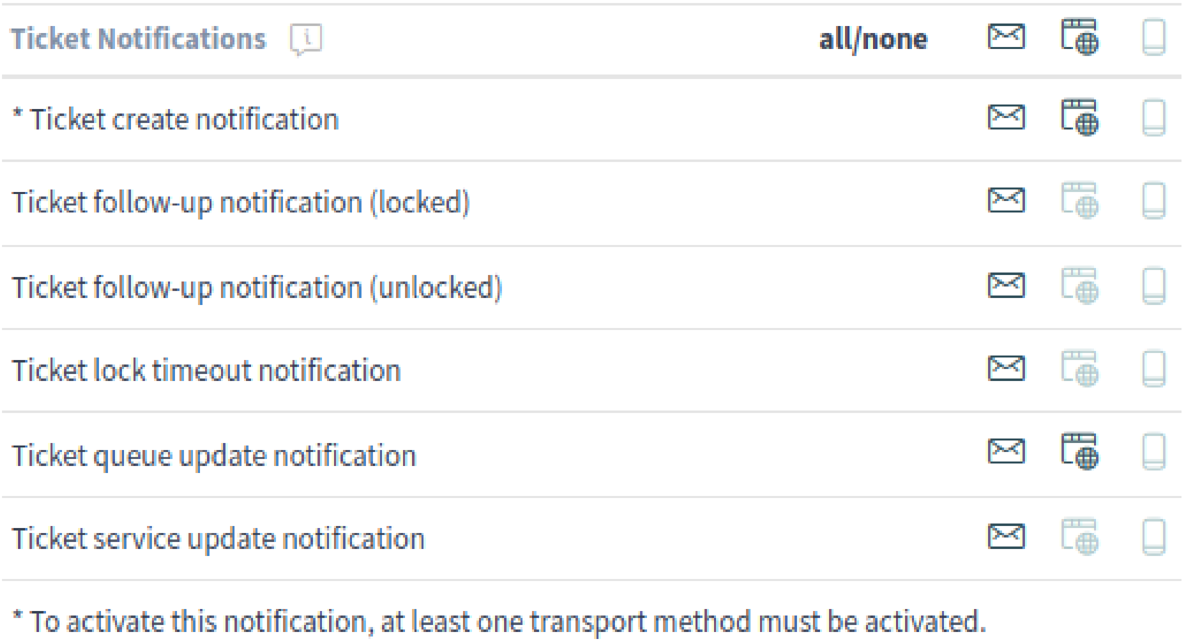
Available for agent in groups The notification will only appear to agents that belong to the selected groups. If no group is selected, it is available for all agents.

Agent preferences tooltip This message will be shown on the agent preferences screen as a tooltip for this notification.

Validity * Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.

Events

Event Here you can choose which events will trigger this notification. An additional ticket filter can be applied below to only send for tickets with certain criteria.



Ticket Notifications | all/none

* Ticket create notification

Ticket follow-up notification (locked)

Ticket follow-up notification (unlocked)

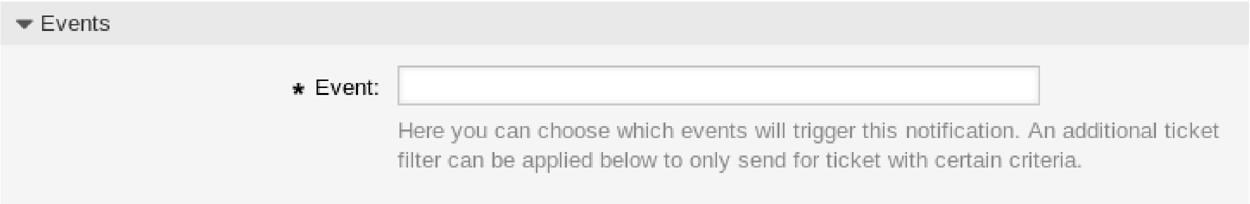
Ticket lock timeout notification

Ticket queue update notification

Ticket service update notification

* To activate this notification, at least one transport method must be activated.

Fig. 62: Personal Ticket Notification Settings

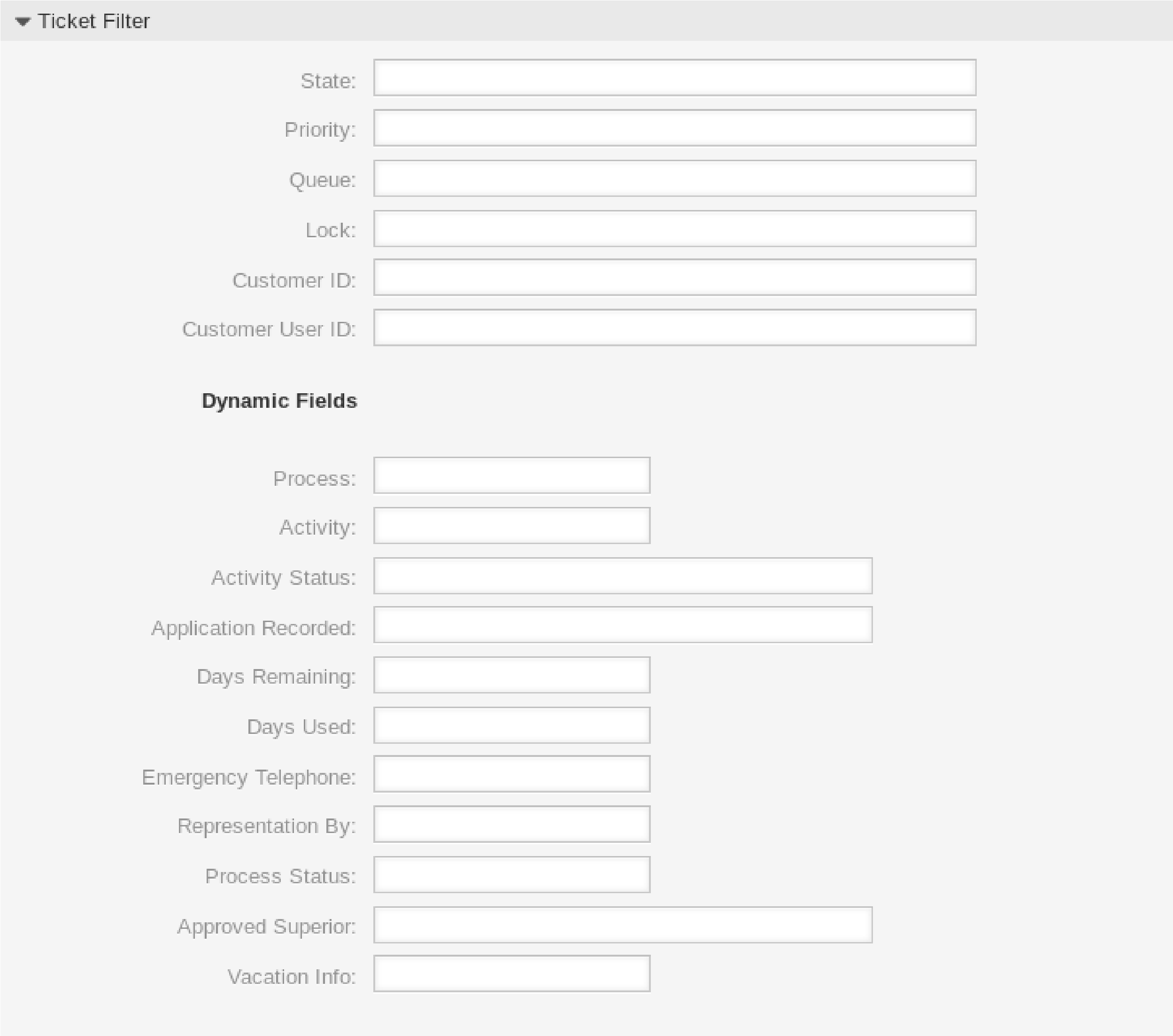


∗ Event:

Here you can choose which events will trigger this notification. An additional ticket filter can be applied below to only send for ticket with certain criteria.

Fig. 63: Ticket Notification Settings - Events

Ticket Filter[1]



▾ Ticket Filter

State:

Priority:

Queue:

Lock:

Customer ID:

Customer User ID:

Dynamic Fields

Process:

Activity:

Activity Status:

Application Recorded:

Days Remaining:

Days Used:

Emergency Telephone:

Representation By:

Process Status:

Approved Superior:

Vacation Info:

Fig. 64: Ticket Notification Settings - Ticket Filter

This widget can optionally be used to narrow the list of tickets by matching configured values:

Note: The values in this list can grow as your system grows. The more *Dynamic Fields* are and features you have in your system, the longer the list will be.

State Filter for a state of the ticket.

Priority Filter for a priority of the ticket.

Queue Filter for a queue in which the ticket is located.

Lock Filter for a lock state of the ticket.

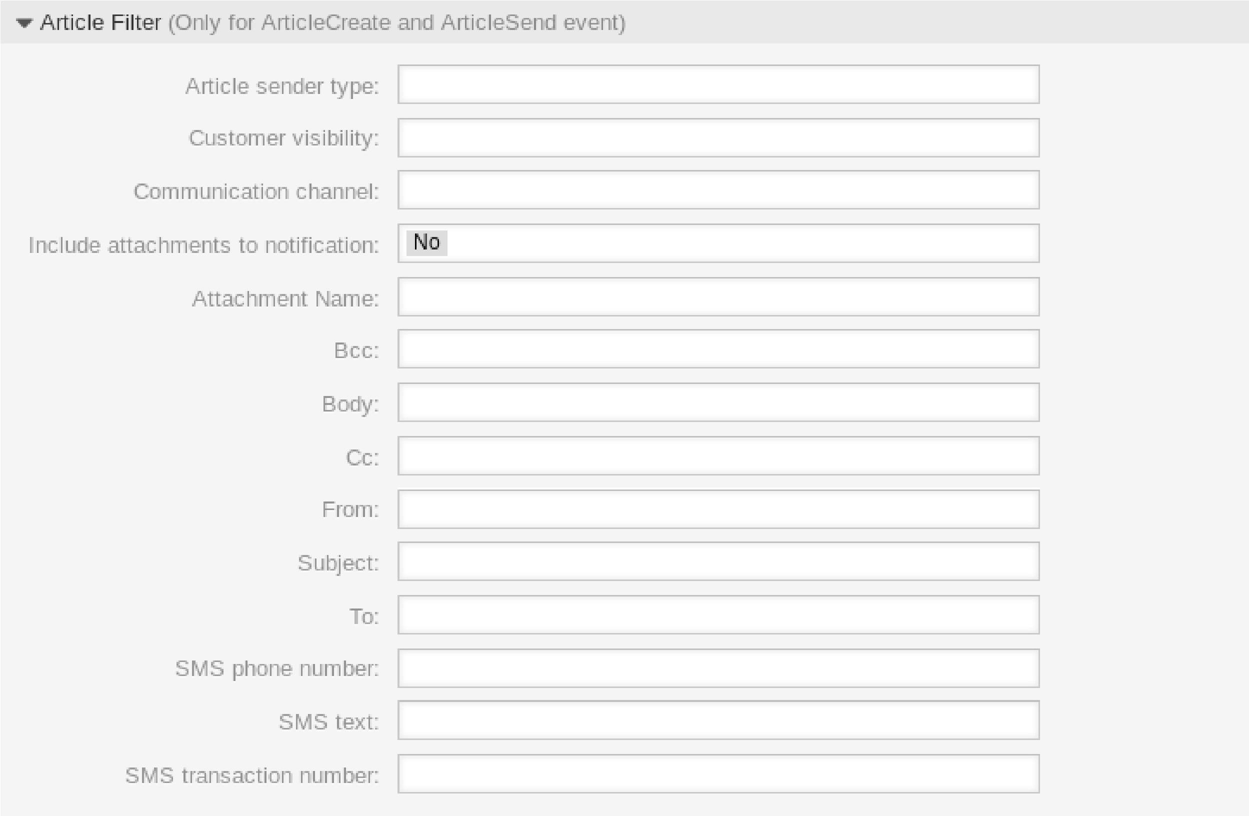
Customer ID Filter for a customer ID of the ticket.

[1] Use of regular expressions as a filter do not work here.

Customer User ID Filter for a customer user ID of the ticket.

Dynamic Fields Filter for some dynamic fields added to the system. For the complete list of dynamic fields see the *Dynamic Fields* chapter.

Article Filter[1]



▾ Article Filter (Only for ArticleCreate and ArticleSend event)

Article sender type:

Customer visibility:

Communication channel:

Include attachments to notification: No

Attachment Name:

Bcc:

Body:

Cc:

From:

Subject:

To:

SMS phone number:

SMS text:

SMS transaction number:

Fig. 65: Ticket Notification Settings - Article Filter

Note: This widget works only if `ArticleCreate` or `ArticleSend` is selected in the *Events* widget.

Article sender type Filter for the sender type of the ticket. Possible values are *agent*, *system* or *customer*.

Customer visibility Filter for the customer visibility. Possible values are *Invisible to customer* or *Visible to customer*.

Communication channel Filter for the communication channel. Possible values are *Chat*, *Email*, *OTRS*, *Phone* or *SMS*.

Include attachments to notification If *Yes* is selected, attachments will be included to notification. Selecting *No* will not use this feature.

Attachment Name Filter for attachment name.

Bcc Filter for blind carbon copy field.

Body Filter for body text.

Cc Filter for carbon copy field.

From Filter for the sender field.

Subject Filter for the subject field.

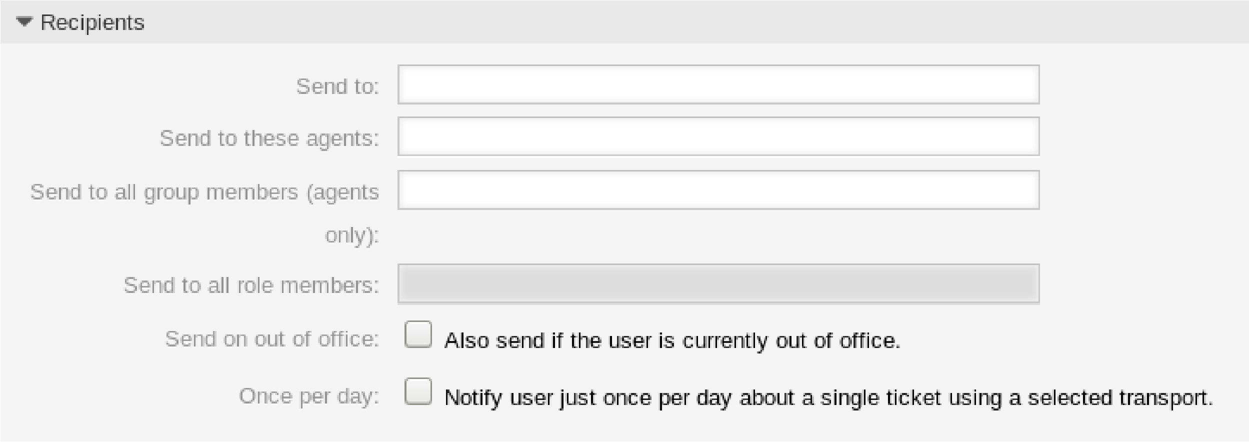
To Filter for the main recipients field.

SMS phone number Filter for an SMS phone number.

SMS text Filter for the SMS text.

SMS transaction number Filter for an SMS transaction number.

Ticket Notification Recipients



▼ Recipients
Send to:
Send to these agents:
Send to all group members (agents only):
Send to all role members:
Send on out of office:
□ Also send if the user is currently out of office.
Once per day:
□ Notify user just once per day about a single ticket using a selected transport.

Fig. 66: Ticket Notification Settings - Recipients

Send to Select which agents should receive the notifications. Possible values are:

- Agent who created the ticket
- Agent who is responsible for the ticket
- Agent who owns the ticket
- All agents subscribed to both the ticket’ s queue and service
- All agents subscribed to the ticket’ s queue
- All agents subscribed to the ticket’ s service
- All agents watching the ticket
- All agents with write permission for the ticket
- All recipients of the first article
- All recipients of the last article
- Customer user of the ticket

Send to these agents One or more agents can be selected who should receive the notifications.

Send to all group members (agents only) One or more groups can be selected whom agents should receive the notifications.

Send to all role members One or more roles can be selected whom agents should receive the notifications.

Send on out of office If this option is checked, the notification will be sent even if the agent is currently out of office.

Once per day Notify users just once per day about a single ticket using a selected transport. If this is the first notification about a ticket, then the notification will be sent. If a notification was already sent before and this option is checked, the OTRS daemon will check the time the last notification was sent. If there was no notification sent in the last 24 hours, the notification will be sent again.

Ticket Notification Methods

Enable this notification method Enable or disable this notification method. A notification method can be email, web view or SMS.

> **Note:** To use the SMS notification method, *Cloud Services* need to be enabled.

Additional recipient email addresses Additional recipients can be added here. Use comma or semicolon to separate the email addresses.

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

Email template Select which email template should be used for the notification.

> **Note:** Additional email templates can be added by placing a `.tt` file into the folder `<OTRS_Home>/Kernel/Output/HTML/Templates/Standard/NotificationEvent/Email/`. See the existing email templates for an example.

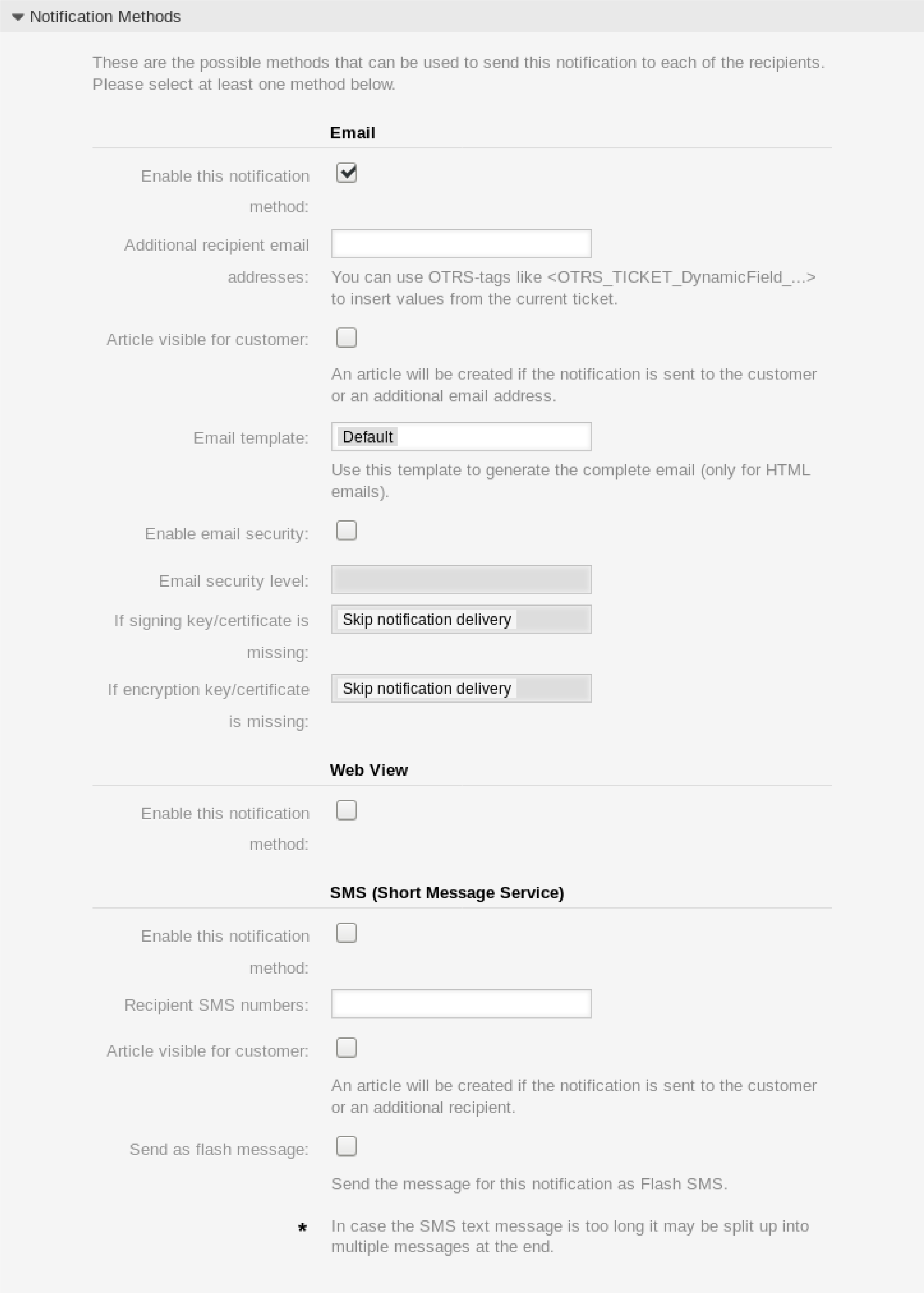
Enable email security Checking this option will encrypt the notification email.

> **Note:** To use this feature, *PGP Keys* or *S/MIME Certificates* need to be enabled.

Email security level If *Enable email security* is checked, then this setting is activated. The following options are available:

- **PGP sign only** Sign only the notification email with PGP key. If no PGP keys have been added to the system, this option is not visible.
- **PGP encrypt only** Encrypt only the notification email with PGP key. If no PGP keys have been added to the system, this option is not visible.
- **PGP sign and encrypt** Sign and encrypt the notification email with PGP key. If no PGP keys have been added to the system, this option is not visible.
- **SMIME sign only** Sign only the notification email with S/MIME certificate. If no S/MIME certificates have been added to the system, this option is not visible.
- **SMIME encrypt only** Encrypt only the notification email with S/MIME certificate. If no S/MIME certificates have been added to the system, this option is not visible.
- **SMIME sign and encrypt** Sign and encrypt the notification email with S/MIME certificate. If no S/MIME certificates have been added to the system, this option is not visible.

> **Note:** To use this feature, *PGP Keys* or *S/MIME Certificates* need to be enabled.



▾ Notification Methods

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

Email

Enable this notification method: ☑

Additional recipient email addresses:

You can use OTRS-tags like <OTRS_TICKET_DynamicField_...> to insert values from the current ticket.

Article visible for customer: □

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

Email template: Default

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

Enable email security: □

Email security level:

If signing key/certificate is missing: Skip notification delivery

If encryption key/certificate is missing: Skip notification delivery

Web View

Enable this notification method: □

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.

Fig. 67: Ticket Notification Settings - Notification Methods

If signing key/certificate is missing Select the method, that should be used if signing key or certificate is missing.

If encryption key/certificate is missing: Select the method, that should be used if encryption key or certificate is missing.

Notification Text

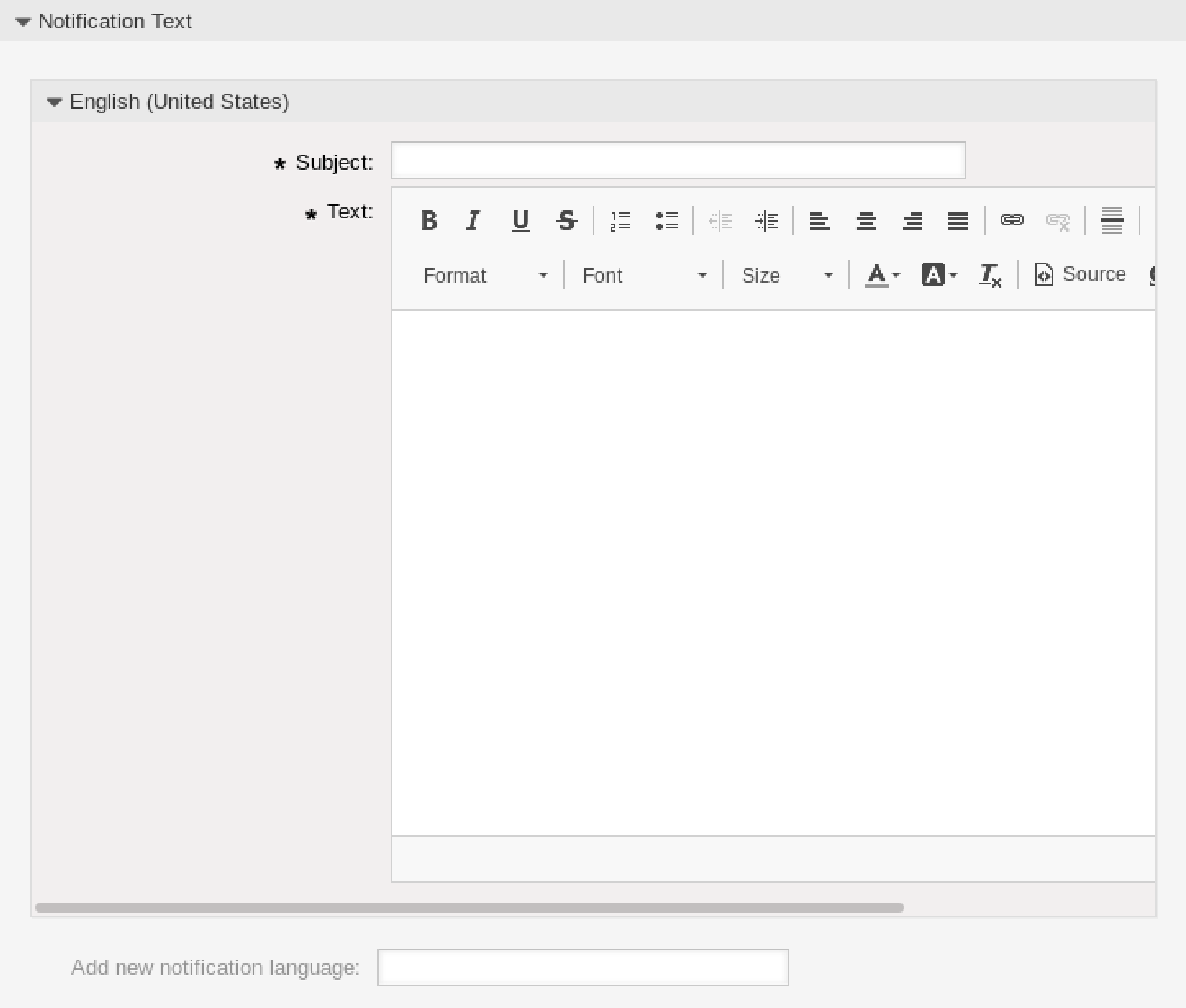


Fig. 68: Ticket Notification Settings - Notification Text

The main content of a notification can be added for each languages with localized subject and body text. It is also possible to define static text content mixed with OTRS smart tags.

Subject * The localized subject for a specific language.

Text * The localized body text for a specific language.

Add new notification language Select which languages should be added to create localized notifications. The language of the customer or agent will be used as found in the customer and agent preferences. Secondarily, the system default language will be chosen. The fall back will always be English.

Warning: Deleting a language from the DefaultUsedLanguages setting that already has a notification text here will make the notification text unusable. If a language is not present or enabled on the system, the corresponding notification text could be deleted if it is not needed anymore.

3.10.3 Ticket Notification Variables

Using variables in the text makes it possible to personalize messages. Variables, known as OTRS tags, are replaced by OTRS when generating the message. Find a list of available tags stems for this resource at the bottom of both add and edit screens.

For example, the variable `<OTRS_TICKET_TicketNumber>` expands to the ticket number allowing a template to include something like the following.

```
Ticket#<OTRS_TICKET_TicketNumber>
```

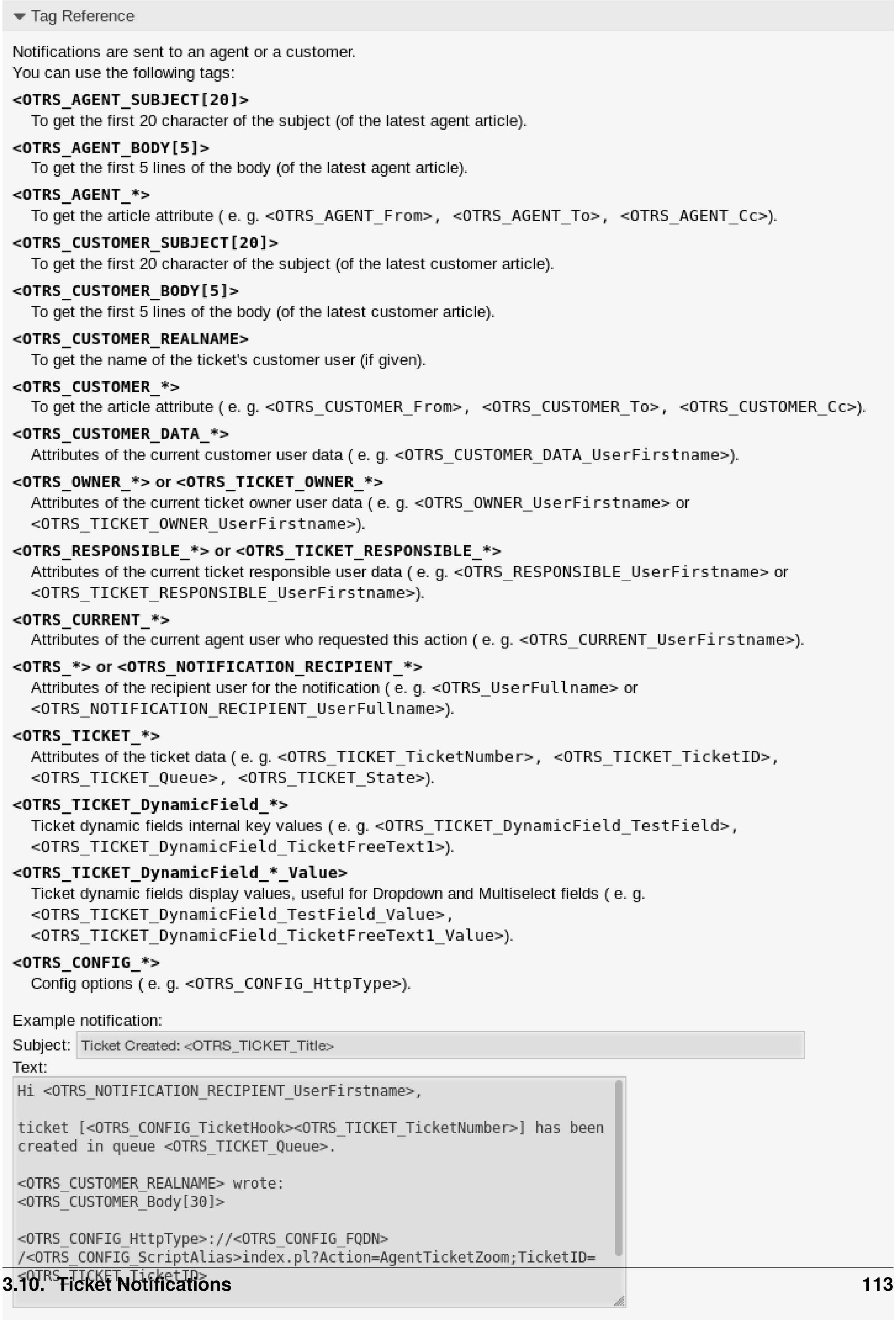
This tag expands, for example to:

```
Ticket#2018101042000012
```

The values of the following variables are translated based on the chosen language of the customer user.

```
OTRSTicket_Type
OTRSTicket_State
OTRSTicket_StateType
OTRSTicket_Lock
OTRSTicket_Priority
```

If the language is not supported the default language is applied.



Notifications are sent to an agent or a customer.
You can use the following tags:

<OTRS_AGENT_SUBJECT[20]>
To get the first 20 character of the subject (of the latest agent article).

<OTRS_AGENT_BODY[5]>
To get the first 5 lines of the body (of the latest agent article).

<OTRS_AGENT_*>
To get the article attribute (e. g. `<OTRS_AGENT_From>, <OTRS_AGENT_To>, <OTRS_AGENT_Cc>`).

<OTRS_CUSTOMER_SUBJECT[20]>
To get the first 20 character of the subject (of the latest customer article).

<OTRS_CUSTOMER_BODY[5]>
To get the first 5 lines of the body (of the latest customer article).

<OTRS_CUSTOMER_REALNAME>
To get the name of the ticket's customer user (if given).

<OTRS_CUSTOMER_*>
To get the article attribute (e. g. `<OTRS_CUSTOMER_From>, <OTRS_CUSTOMER_To>, <OTRS_CUSTOMER_Cc>`).

<OTRS_CUSTOMER_DATA_*>
Attributes of the current customer user data (e. g. `<OTRS_CUSTOMER_DATA_UserFirstname>`).

<OTRS_OWNER_*> or <OTRS_TICKET_OWNER_*>
Attributes of the current ticket owner user data (e. g. `<OTRS_OWNER_UserFirstname>` or `<OTRS_TICKET_OWNER_UserFirstname>`).

<OTRS_RESPONSIBLE_*> or <OTRS_TICKET_RESPONSIBLE_*>
Attributes of the current ticket responsible user data (e. g. `<OTRS_RESPONSIBLE_UserFirstname>` or `<OTRS_TICKET_RESPONSIBLE_UserFirstname>`).

<OTRS_CURRENT_*>
Attributes of the current agent user who requested this action (e. g. `<OTRS_CURRENT_UserFirstname>`).

<OTRS_*> or <OTRS_NOTIFICATION_RECIPIENT_*>
Attributes of the recipient user for the notification (e. g. `<OTRS_UserFullname>` or `<OTRS_NOTIFICATION_RECIPIENT_UserFullname>`).

<OTRS_TICKET_*>
Attributes of the ticket data (e. g. `<OTRS_TICKET_TicketNumber>, <OTRS_TICKET_TicketID>, <OTRS_TICKET_Queue>, <OTRS_TICKET_State>`).

<OTRS_TICKET_DynamicField_*>
Ticket dynamic fields internal key values (e. g. `<OTRS_TICKET_DynamicField_TestField>, <OTRS_TICKET_DynamicField_TicketFreeText1>`).

<OTRS_TICKET_DynamicField_*_Value>
Ticket dynamic fields display values, useful for Dropdown and Multiselect fields (e. g. `<OTRS_TICKET_DynamicField_TestField_Value>, <OTRS_TICKET_DynamicField_TicketFreeText1_Value>`).

<OTRS_CONFIG_*>
Config options (e. g. `<OTRS_CONFIG_HttpType>`).

Example notification:

Subject: Ticket Created: <OTRS_TICKET_Title>

Text:

```
Hi <OTRS_NOTIFICATION_RECIPIENT_UserFirstname>,

ticket [<OTRS_CONFIG_TicketHook><OTRS_TICKET_TicketNumber>] has been
created in queue <OTRS_TICKET_Queue>.

<OTRS_CUSTOMER_REALNAME> wrote:
<OTRS_CUSTOMER_Body[30]>

<OTRS_CONFIG_HttpType>://<OTRS_CONFIG_FQDN>
/<OTRS_CONFIG_ScriptAlias>index.pl?Action=AgentTicketZoom;TicketID=
<OTRS_TICKET_TicketID>
```

Fig. 69: Ticket Notification Variables

CHAPTER 4

Users, Groups & Roles

Simple and complex organizations need a flexible way to control access to their service desk. Access, resources, and permissions must be orchestrated so that, the users and customers have access to the resources they need and their data is protected as needed by the system.

OTRS provides powerful tools for this purpose and their use is described in the following chapter.

4.1 Agents

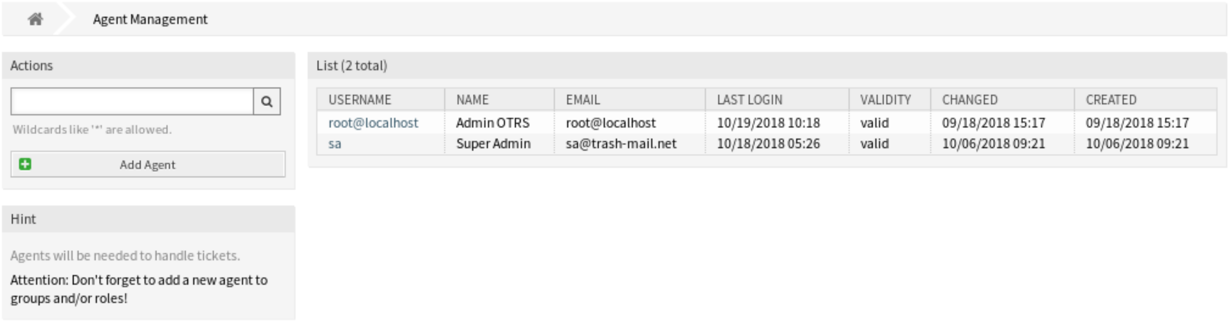
Agent and access management for your service desk should be easy. Flexibility in adding, editing, invalidating access and a quick overview of which permissions a user has will help you maintain a clean permissions system and record of your setup in OTRS.

OTRS aids you giving you the power to manage agents within OTRS across multiple back ends. OTRS can use up-to ten back end sources, even marking some as read-only. Managing user settings centrally, an administrator can quickly invalidate a compromised account or set an account to out-of-office in case of an unexpected illness.

Use this screen to add agents to the system. A fresh OTRS installation contains an agent with administrator privileges by default. The agent management screen is available in the *Agents* module of the *Users, Groups & Roles* group.

Warning: The superuser account username is *root@localhost*. Don’t use the superuser account to work with OTRS! Create new agents and work with these accounts instead. One of the adverse effects is that *Access Control Lists (ACL)* will not have an effect on this user.

4.1.1 Manage Agents



Wildcards like '*' are allowed.

List (2 total)

USERNAME	NAME	EMAIL	LAST LOGIN	VALIDITY	CHANGED	CREATED
root@localhost	Admin OTRS	root@localhost	10/19/2018 10:18	valid	09/18/2018 15:17	09/18/2018 15:17
sa	Super Admin	sa@trash-mail.net	10/18/2018 05:26	valid	10/06/2018 09:21	10/06/2018 09:21

Agents will be needed to handle tickets.

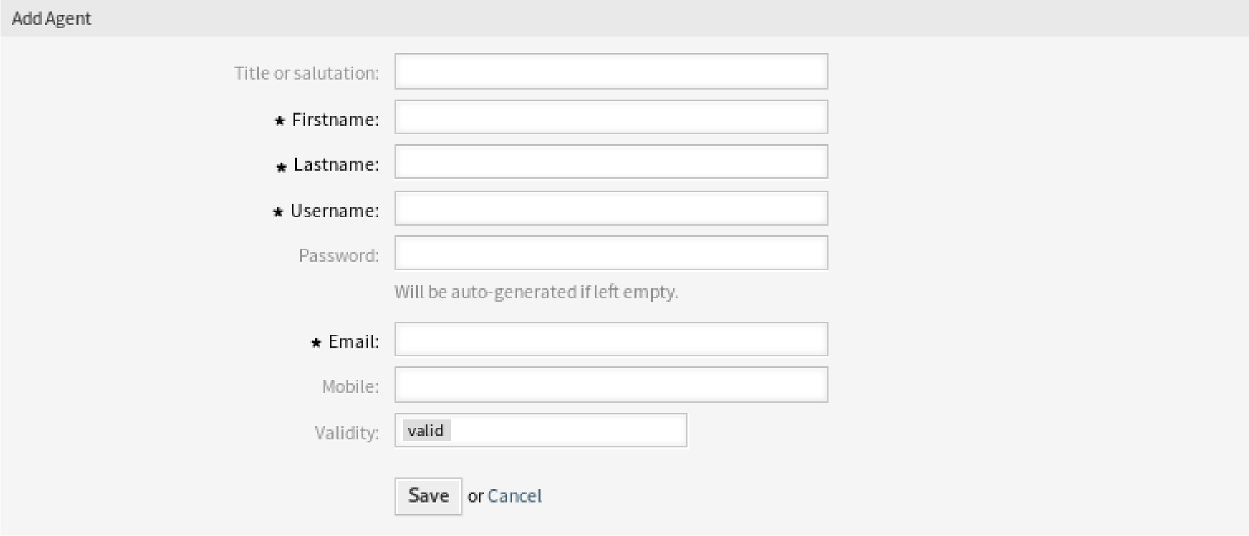
Attention: Don't forget to add a new agent to groups and/or roles!

Fig. 1: Agent Management Screen

Note: Adding or editing an agent is possible only by using database back end. Using explicitly external directory services like LDAP and, based on configuration, some databases are read-only. Personal preferences like out-of-office can still be set.

To add an agent:

1. Click on the *Add Agent* button in the left sidebar.
2. Fill in the required fields.
3. Click on the *Save* button.



Title or salutation:	
∗ Firstname:	
∗ Lastname:	
∗ Username:	
Password:	Will be auto-generated if left empty.
∗ Email:	
Mobile:	
Validity:	valid

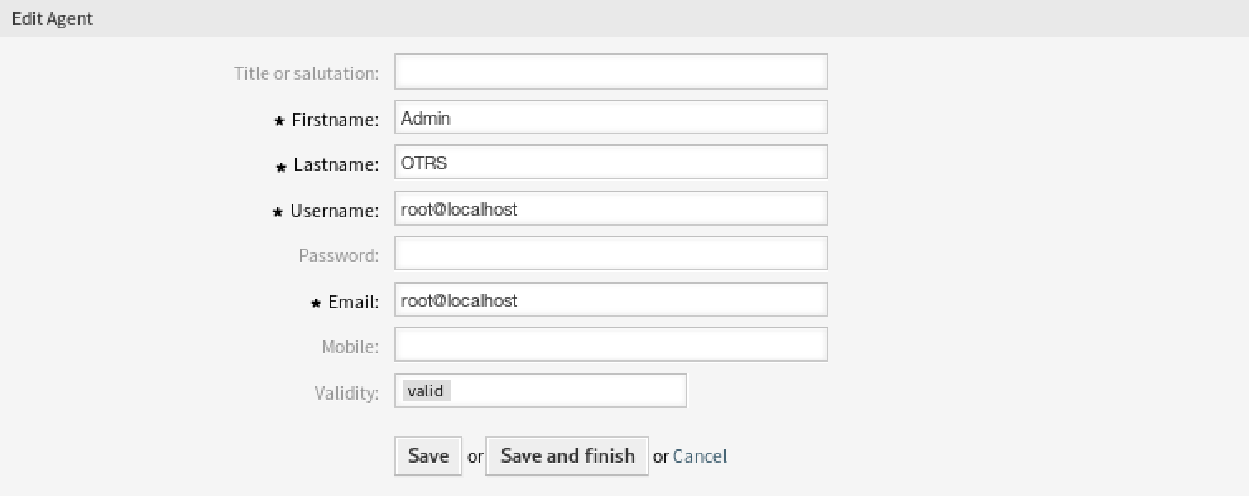
Fig. 2: Add Agent Screen

Warning: Agents can not be deleted from the system. They can only be deactivated by setting the *Validity* option to *invalid* or *invalid-temporarily*.

To edit an agent:

1. Click on an agent in the list of agents.

2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.



Edit Agent

Title or salutation:	
★ Firstname:	Admin
★ Lastname:	OTRS
★ Username:	root@localhost
Password:	
★ Email:	root@localhost
Mobile:	
Validity:	valid

Fig. 3: Edit Agent Screen

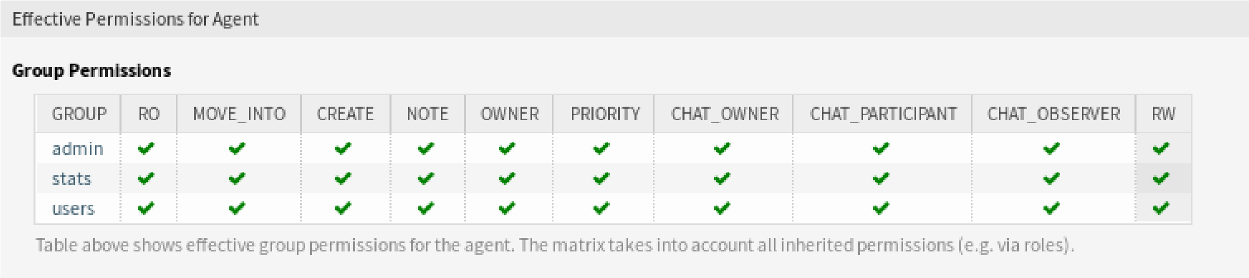
It is also possible to edit the agent personal preferences. To do this, click on the *Edit personal preferences for this agent* button in the left sidebar of the *Edit Agent* screen.

To find an agent:

1. Enter a search term to the search box in the left sidebar.
2. Click on the magnifying glass icon in the right part of the field or hit an `Enter`.

Note: If several agents are added to the system, use the search box to find a particular agent. Only the first 1000 agents are listed by default.

The agent permissions can be controlled by adding an agent to *Groups* or *Roles*. This can result a complex matrix of permissions. The effective permissions for an agent can be checked in the bottom of the *Edit Agent* screen. If roles (recommended) are used, this screen will reflect the combined permissions as dictated by the roles.



Effective Permissions for Agent

Group Permissions

GROUP	RO	MOVE_INTO	CREATE	NOTE	OWNER	PRIORITY	CHAT_OWNER	CHAT_PARTICIPANT	CHAT_OBSERVER	RW
admin	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
stats	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
users	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table above shows effective group permissions for the agent. The matrix takes into account all inherited permissions (e.g. via roles).

Fig. 4: Effective Permissions for Agent Widget

4.1.2 Agent Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Note: These are the default fields available for the internal database table.

Title or salutation Some name prefix can be added here like *Mr.*, *Dr.*, *Jr.*, etc.

Firstname * The first name of the agent.

Lastname * The last name of the agent.

> **See also:**
>
> The agent display name can be set via the system configuration setting `FirstnameLastnameOrder`.

Username * The username of the agent to login to the system.

Password The password of the agent. This will be auto-generated if left empty.

Email * The email address of the agent.

> **Note:** The email syntax and validity of an available MX record could prevent you from submitting this form. For some systems it may be acceptable to turn off these checks. See the *Agent Configuration Options*.

Mobile The mobile phone number of the agent.

Validity * Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.

4.1.3 Agent Back End Sources

Agents can be read and synchronized from an Active Directory® or LDAP server.

In case of using *On-Premise* system or *((OTRS)) Community Edition*, you may use the following section to configure your system.

Note: The following configurations are taken care of by the *Customer Solution Team* in **OTRS**. Please contact the *Customer Solution Team* via support@otrs.com or in the OTRS Portal.

Agent Authentication Back End

The users table houses your agent data per default. You may attach a directory server (LDAP/Active Directory) to your installation for authentication. To achieve this copy the following block from the `Defaults.pm` and paste it into the `Config.pm`.

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

(continues on next page)

(continued from previous page)

```
#    $Self->{'AuthModule::LDAP::UID'} = 'uid';

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

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

# in case you want to add always one filter to each ldap query, use
# this option. e. g. AlwaysFilter => '(mail=*)' or AlwaysFilter =>
↪'(objectclass=user)'
# or if you want to filter with a locigal OR-Expression, like AlwaysFilter =>
↪'(|(mail=*abc.com)(mail=*xyz.com))'
#    $Self->{'AuthModule::LDAP::AlwaysFilter'} = '';

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

# In case you want to convert all given usernames to lower letters you
# should activate this option. It might be helpful if databases are
# in use that do not distinguish selects for upper and lower case letters
# (Oracle, postgresql). User might be synched twice, if this option
# is not in use.
#    $Self->{'AuthModule::LDAP::UserLowerCase'} = 0;

# In case you need to use OTRS in iso-charset, you can define this
# by using this option (converts utf-8 data from LDAP to iso).
#    $Self->{'AuthModule::LDAP::Charset'} = 'iso-8859-1';

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

# Die if backend can't work, e. g. can't connect to server.
#    $Self->{'AuthModule::LDAP::Die'} = 1;

You should uncomment the following settings as a minimum.
```

The minimum required to connect to a directory server is:

```
$Self->{AuthModule} = 'Kernel::System::Auth::LDAP';
$Self->{'AuthModule::LDAP::Host'} = 'ldap.example.com';
$Self->{'AuthModule::LDAP::BaseDN'} = 'dc=example,dc=com';
$Self->{'AuthModule::LDAP::UID'} = 'uid';
```

Host The DNS name or IP of your directory server.

BaseDN The starting point in your directory tree.

UID The attribute used for login and identification.

> **Note:** This is `sAMAccountName` for an Active Directory.

To use multiple back ends, add an additional section of the example code to the `Config.pm`. Please make sure to add a numeric value [1-9] to all settings to indicate which settings belong to which back end.

```
### Backend One
$Self->{AuthModule} = 'Kernel::System::Auth::LDAP';
$Self->{'AuthModule::LDAP::Host'} = 'ldap.example.com';
$Self->{'AuthModule::LDAP::BaseDN'} = 'dc=example,dc=com';
$Self->{'AuthModule::LDAP::UID'} = 'uid';

### Backend Two
$Self->{AuthModule1} = 'Kernel::System::Auth::LDAP';
$Self->{'AuthModule::LDAP::Host1'} = 'ldap.example.com';
$Self->{'AuthModule::LDAP::BaseDN1'} = 'dc=example,dc=com';
$Self->{'AuthModule::LDAP::UID1'} = 'uid';
```

> **Warning:** All back ends will are used in succession. The UID must be unique to all back ends, otherwise some side effects may occur.

To synchronize with a specific directory server (see *Agent User Data* below), you must add the appropriate setting to your *Agent Authentication Back End*. To achieve this copy the following block from the `Defaults.pm` and paste it into the `Config.pm`.

```
$Self->{'AuthModule::UseSyncBackend'} = 'AuthSyncBackend';
```

To use multiple back ends, add an additional section of the example code to the `Config.pm`. Please make sure to add a numeric value [1-9] to all settings to indicate which settings belong to which back end.

```
$Self->{'AuthModule::UseSyncBackend1'} = 'AuthSyncBackend1';
```

Reuse of an *Agent Synchronization Back End* is also possible.

```
$Self->{'AuthModule::UseSyncBackend1'} = 'AuthSyncBackend';
```

Agent Synchronization Back End

It is advisable to synchronize the agent data so that agents need not be manually added to the users table prior to authorization. Additionally, groups and roles can be added automatically using security objects of the directory server.

Agent User Data

Syncing user data upon login. To achieve this copy the following block from the `Defaults.pm` and paste it into the `Config.pm`.

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

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

# in case you want to add always one filter to each ldap query, use
# this option. e. g. AlwaysFilter => '(mail=*)' or AlwaysFilter =>
↪'(objectclass=user)'
# or if you want to filter with a logical OR-Expression, like AlwaysFilter =>
↪'(|(mail=*abc.com)(mail=*xyz.com))'
#    $Self->{'AuthSyncModule::LDAP::AlwaysFilter'} = '';

# AuthSyncModule::LDAP::UserSyncMap
# (map if agent should create/synced from LDAP to DB after successful login)
# you may specify LDAP-Fields as either
#   * list, which will check each field. first existing will be picked ( [
↪"givenName","cn","_empty"] )
#   * name of an LDAP-Field (may return empty strings) ("givenName")
#   * fixed strings, prefixed with an underscore: "_test", which will always␣
↪return this fixed string
#    $Self->{'AuthSyncModule::LDAP::UserSyncMap'} = {
#        # DB -> LDAP
#        UserFirstname => 'givenName',
#        UserLastname  => 'sn',
#        UserEmail     => 'mail',
#    };
```

The minimum required to connect to a directory server is:

```
$Self->{AuthSyncModule} = 'Kernel::System::Auth::Sync::LDAP';
$Self->{'AuthSyncModule::LDAP::Host'} = 'ldap.example.com';
$Self->{'AuthSyncModule::LDAP::BaseDN'} = 'dc=example,dc=com';
$Self->{'AuthSyncModule::LDAP::UID'} = 'uid';
```

Host The DNS name or IP of your directory server.

BaseDN The starting point in your directory tree.

UID The attribute used for login and identification.

> **Note:** This is `sAMAccountName` for an Active Directory.

Note: Multiple *Agent Synchronization Back End* blocks can be used. Please make sure to add a numeric value [1-9] to all settings to indicate which settings belong to which back end. Each `AuthSyncModule` must be explicitly used in an *Agent Authentication Back End*.

Agent Group Data

It is possible to use security objects to synchronize users to OTRS groups. To achieve this copy the following block from the `Defaults.pm` and paste it into the `Config.pm`.

Note: *Groups* must be available in OTRS to use this feature.

Agent Role Data

It is possible to use security objects to synchronize users to OTRS roles. To achieve this copy the following block from the `Defaults.pm` and paste it into the `Config.pm`.

Note: *Roles* must be available in OTRS to use this feature.

4.1.4 Agent Configuration Options

4.2 Agents Groups

Efficient and straightforward management of permissions is essential in a growing business. Easy assignment of a particular user to a group for quick access, or to remove access, to resources is a must in every case.

The OTRS interface provides you both with the possibility to manage an agent’ s access to one or more particular groups. As well, you can change multiple users access to any one group, efficiently and elegantly.

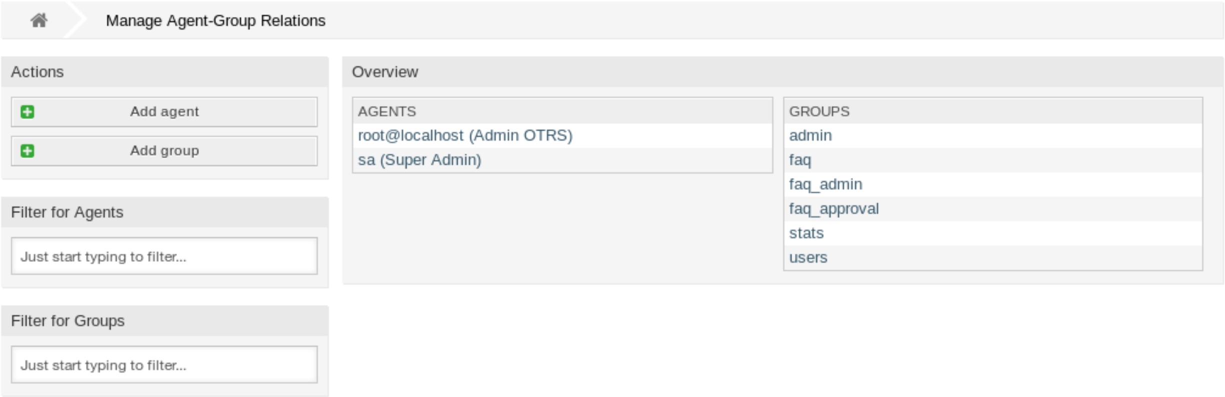
Use this screen to add one or more agents to one or more groups. To use this function, at least one agent and one group need to have been added to the system. The management screen is available in the *Agents Groups* module of the *Users, Groups & Roles* group.

4.2.1 Manage Agents Groups Relations

To assign some groups to an agent:

1. Click on an agent in the *Agents* column.
2. Select the permissions you would like to connect the agent to groups with.
3. Click on the *Save* or *Save and finish* button.

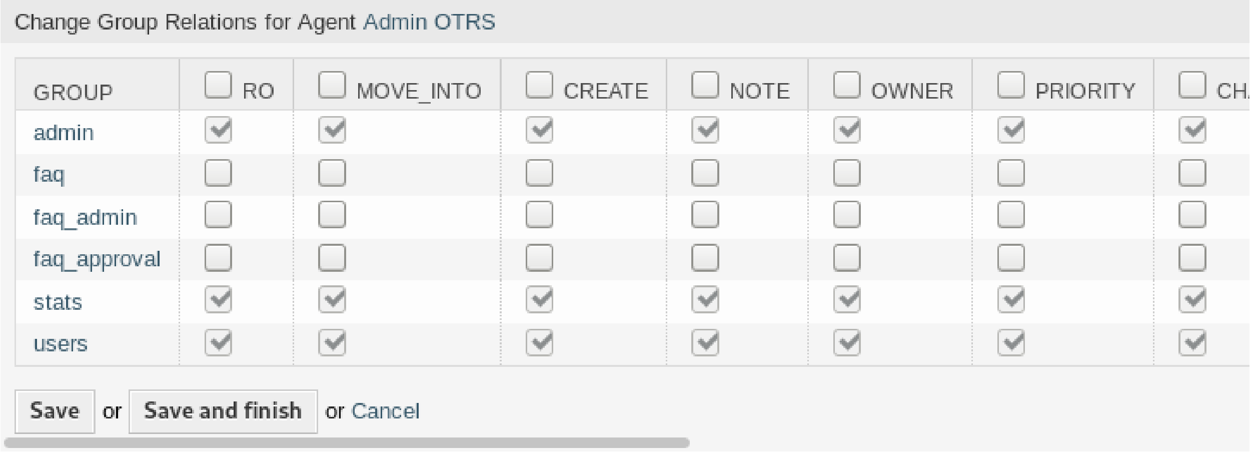
To assign some agents to a group:


Manage Agent-Group Relations
Actions
Add agent
Add group
Filter for Agents
Just start typing to filter...
Filter for Groups
Just start typing to filter...
Overview

AGENTS
root@localhost (Admin OTRS)
sa (Super Admin)

GROUPS
admin
faq
faq_admin
faq_approval
stats
users

Fig. 5: Manage Agent-Group Relations

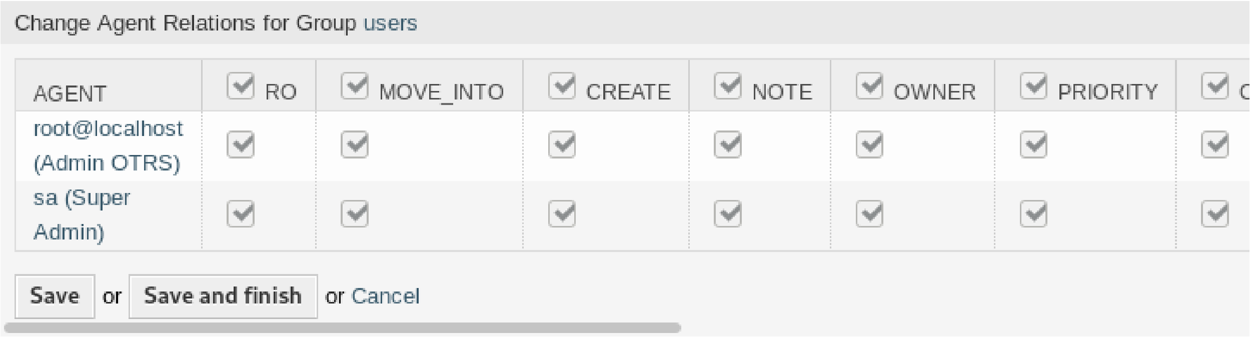


Change Group Relations for Agent Admin OTRS

GROUP	□ RO	□ MOVE_INTO	□ CREATE	□ NOTE	□ OWNER	□ PRIORITY	□ CH
admin	☑	☑	☑	☑	☑	☑	☑
faq	□	□	□	□	□	□	□
faq_admin	□	□	□	□	□	□	□
faq_approval	□	□	□	□	□	□	□
stats	☑	☑	☑	☑	☑	☑	☑
users	☑	☑	☑	☑	☑	☑	☑

Fig. 6: Change Group Relations for Agent

1. Click on a group in the *Groups* column.
2. Select the permissions you would like to connect the group to agents with.
3. Click on the *Save* or *Save and finish* button.



Change Agent Relations for Group users

AGENT	☑ RO	☑ MOVE_INTO	☑ CREATE	☑ NOTE	☑ OWNER	☑ PRIORITY	☑ C
root@localhost (Admin OTRS)	☑	☑	☑	☑	☑	☑	☑
sa (Super Admin)	☑	☑	☑	☑	☑	☑	☑

Save or Save and finish or Cancel

Fig. 7: Change Agent Relations for Group

Note: If several agents or groups are added to the system, use the filter box to find a particular agent or group by just typing the name to filter.

Multiple agents or groups can be assigned in both screens at the same time. Additionally clicking on an agent or clicking on a group in the relations screen will open the *Edit Agent* screen or the *Edit Group* screen accordingly.

Warning: Accessing an agent or a group provides no back link to the relations screen.

4.2.2 Agents Groups Relations Reference

When assigning an agent to a group or vice versa, several permissions can be set as connection between an agent and a group. The following permissions are available by default:

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

move_into Permissions to move tickets into this group/queue.

create Permissions to create tickets in this group/queue.

note Permissions to add notes to tickets in this group/queue.

owner Permissions to change the owner of tickets in this group/queue.

priority Permissions to change the ticket priority in this group/queue.

chat_owner Agents have full rights for a chat and can accept chat requests.

chat_participant Agents may normally participate in a chat.

chat_observer Agents may take part silently in a chat.

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

See also:

Not all available permissions are shown by default. See System::Permission setting for permissions that can be added. These additional permissions can be added:

stats Gives access to the stats page.

bounce Permissions to bounce an email message (with bounce button in *Ticket Zoom* screen).

compose Permissions to compose an answer for a ticket.

customer Permissions to change the customer of a ticket.

forward Permissions to forward a message (with the forward button).

pending Permissions to set a ticket to pending.

phone Permissions to add a phone call to a ticket.

responsible Permissions to change the responsible agent for a ticket.

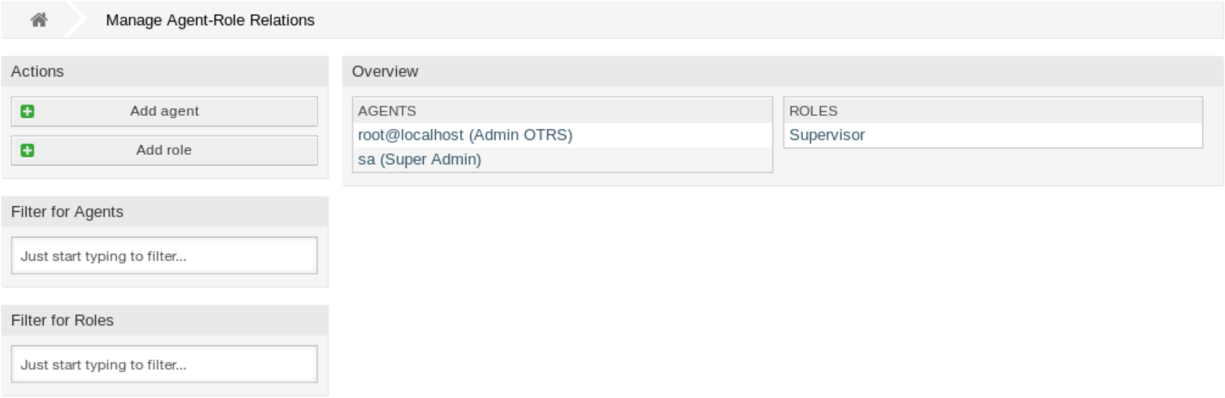
Note: By setting a checkbox in the header of a column will set all the checkboxes in the selected column. By setting the checkbox in the last *rw* column will set all the checkboxes in the selected row.

4.3 Agents Roles

As an organization grows, groups cannot be the denominator used for processing access rights. Roles become more and more a need because a role has a special set of permissions assigned. One mustn’t give individual permissions, but the role carries the permissions built-in.

OTRS allows easy access to a predefined set of permissions via one or more roles defined. These roles are assigned easily to one or more agents, or one or more agent to a role.

Use this screen to add one or more agents to one or more roles. To use this function, at least one agent and one role need to have been added to the system. The management screen is available in the *Agents Roles* module of the *Users, Groups & Roles* group.


Manage Agent-Role Relations
Actions
Add agent
Add role
Filter for Agents
Just start typing to filter...
Filter for Roles
Just start typing to filter...
Overview
AGENTS
ROLES

AGENTS	ROLES
root@localhost (Admin OTRS)	Supervisor
sa (Super Admin)	

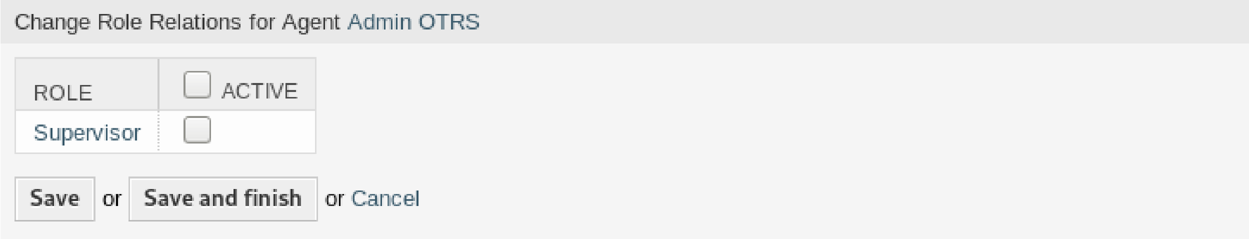
Fig. 8: Manage Agent-Role Relations

4.3.1 Manage Agents Roles Relations

To assign some roles to an agent:

1. Click on an agent in the *Agents* column.

2. Select the roles you would like to the agent belongs to.
3. Click on the *Save* or *Save and finish* button.



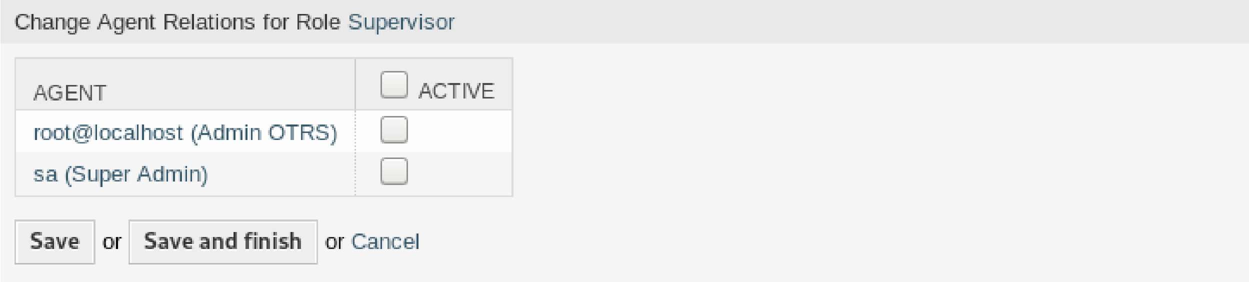
Change Role Relations for Agent Admin OTRS

ROLE	□ ACTIVE
Supervisor	□

Fig. 9: Change Role Relations for Agent

To assign some agents to a role:

1. Click on a role in the *Roles* column.
2. Select the agents you would like to add to the role.
3. Click on the *Save* or *Save and finish* button.



Change Agent Relations for Role Supervisor

AGENT	□ ACTIVE
root@localhost (Admin OTRS)	□
sa (Super Admin)	□

Fig. 10: Change Agent Relations for Role

Note: If several agents or roles are added to the system, use the filter box to find a particular agent or role by just typing the name to filter.

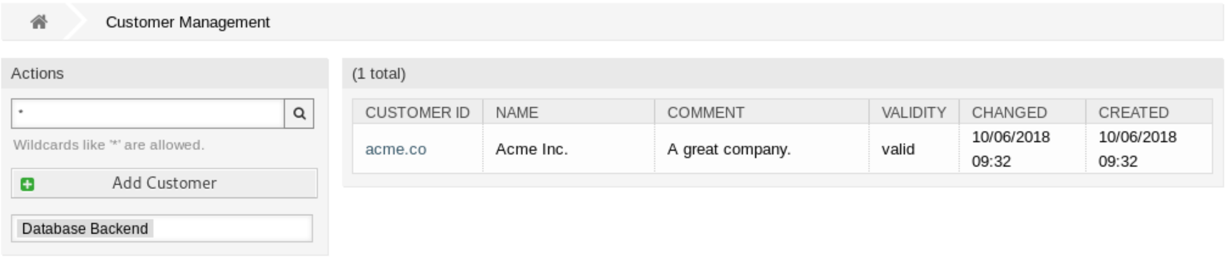
Multiple agents or roles can be assigned in both screens at the same time. Additionally clicking on an agent or clicking on a role in the relations screen will open the *Edit Agent* screen or the *Edit Role* screen accordingly.

Warning: Accessing an agent or a role provides no back link to the relations screen.

Note: By setting the checkbox in the header of the column *Active* will set all the checkboxes in the column.

4.4 Customers

Use this screen to add customer companies to the system. A fresh OTRS installation contains no customers by default. The customer management screen is available in the *Customers* module of the *Users, Groups & Roles* group.



Customer Management

Actions

*

Wildcards like '*' are allowed.

Add Customer

Database Backend

(1 total)

CUSTOMER ID	NAME	COMMENT	VALIDITY	CHANGED	CREATED
acme.co	Acme Inc.	A great company.	valid	10/06/2018 09:32	10/06/2018 09:32

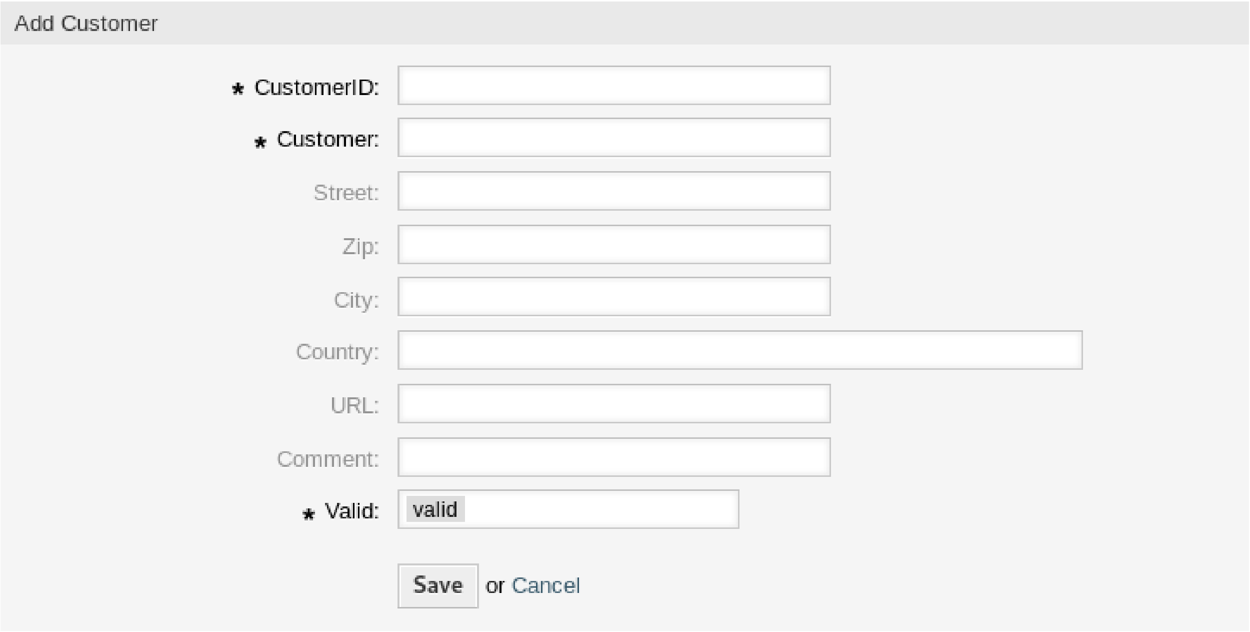
Fig. 11: Customer Management Screen

4.4.1 Manage Customers

Note: Adding or editing a customer is possible only by using database back end. Using external directory services like LDAP will disable the customer management functionality.

To add a customer:

1. Click on the *Add Customer* button in the left sidebar.
2. Fill in the required fields.
3. Click on the *Save* button.



Add Customer

* CustomerID:

* Customer:

Street:

Zip:

City:

Country:

URL:

Comment:

* Valid: valid

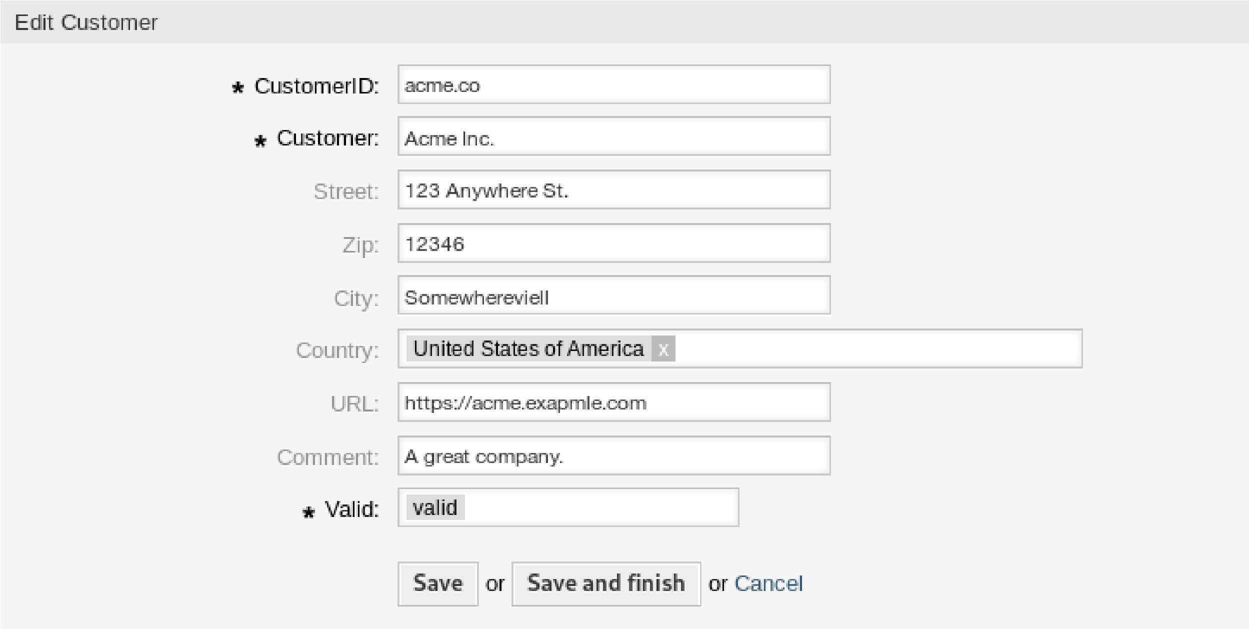
Save or Cancel

Fig. 12: Add Customer Screen

Warning: Customers can not be deleted from the system. They can only be deactivated by setting the *Validity* option to *invalid* or *invalid-temporarily*.

To edit a customer:

1. Click on a customer in the list of customers.
2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.



Edit Customer

* CustomerID:	acme.co
* Customer:	Acme Inc.
Street:	123 Anywhere St.
Zip:	12346
City:	Somewhereviell
Country:	United States of America
URL:	https://acme.exapmle.com
Comment:	A great company.
* Valid:	valid

Fig. 13: Edit Customer Screen

To find a customer:

1. Enter a search term to the search box in the left sidebar.
2. Click on the magnifying glass icon in the right part of the field or hit an `Enter`.

Note: If several customers are added to the system, use the search box to find a particular customer. Only the first 1000 customers are listed by default.

4.4.2 Customer Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Customer ID * The internal name of the customer. Should contain only letters, numbers and some special characters.

Customer * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Street The street name of the customer.

Zip The zip code of the customer.

City The headquarter city of the customer.

Country The country of the customer. Choose a country from the list.

URL The web page or other URL of the customer.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

Validity * Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.

4.5 Customers   Groups

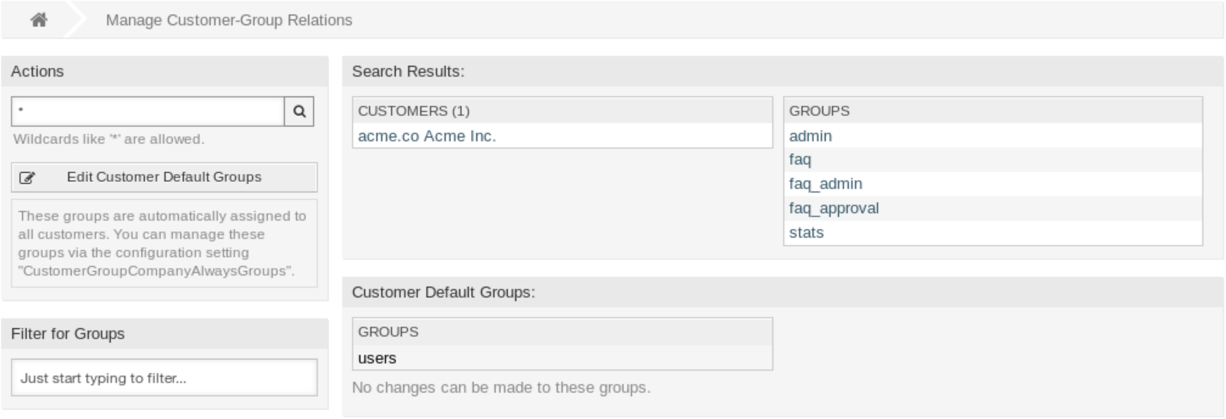
Your organization grows, and it’ s not practical at some point to assign permissions to individual users, you need to assign the permissions to all customer users of a customer.

OTRS allows you to assign *group* permissions to a *customer*. Access works just the same as for agents, preventing a customer from modifying and viewing a request. Thus allowing the customer to focus on the results of the original communication and funneling the discussion through one ticket.

See also:

Assign a single customer user to a group using *Customer Users   Groups*.

Use this screen to add one or more customers to one or more groups. To use this function, at least one customer and one group need to have been added to the system. The management screen is available in the *Customers   Groups* module of the *Users, Groups & Roles* group.



Wildcards like '*' are allowed.

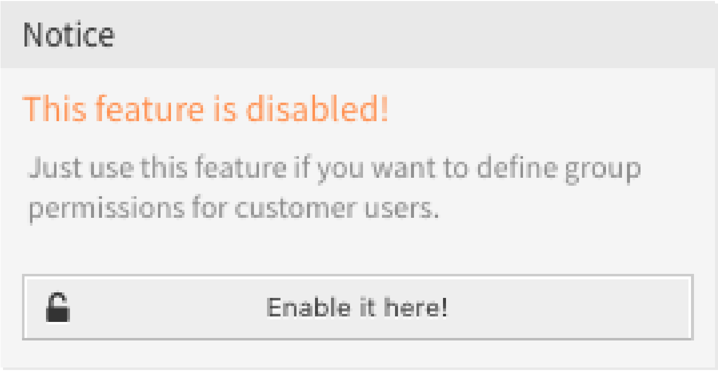
These groups are automatically assigned to all customers. You can manage these groups via the configuration setting "CustomerGroupCompanyAlwaysGroups".

No changes can be made to these groups.

Fig. 14: Manage Customer-Group Relations

Customer group support needs to be enabled in at least one customer user *back end* to use this function. For the default OTRS *back end*, this can be enabled in the system configuration by clicking on the *Enable it here!* button.

Note: To enable this feature in systems using a directory server or multiple non-default back ends, a custom configuration file needs to be placed in `Kernel/Config/Files` (for example named `ZZZ_CustomerBackend.pm`). Once activated, all customer users from this back end will require group assignment.



Notice

This feature is disabled!

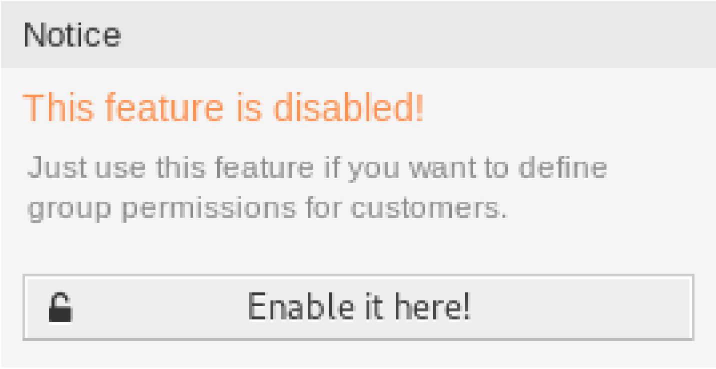
Just use this feature if you want to define group permissions for customer users.

Fig. 15: Enable Customer Group Feature

> **Warning:** After making changes to the back end, the server cache will be deleted, which may cause a temporary drop in performance.

4.5.1 Manage Customers Groups Relations

Note: To be able to use this feature, you have to activate its setting first.



Notice

This feature is disabled!

Just use this feature if you want to define group permissions for customers.

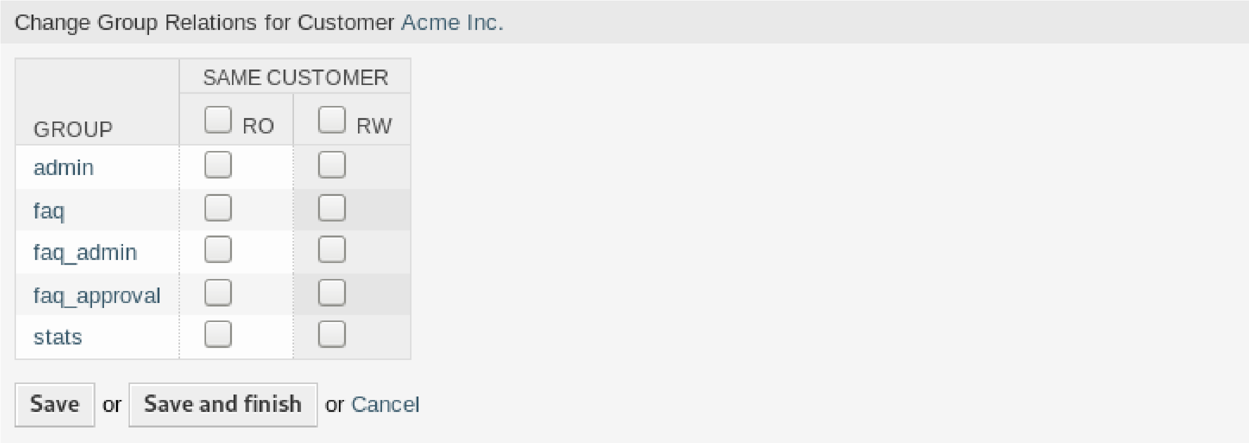
Fig. 16: Enable Customer-Group Support

To assign some groups to a customer:

1. Click on a customer in the *Customers* column.
2. Select the permissions you would like to connect the customer to groups with.
3. Click on the *Save* or *Save and finish* button.

To assign some customers to a group:

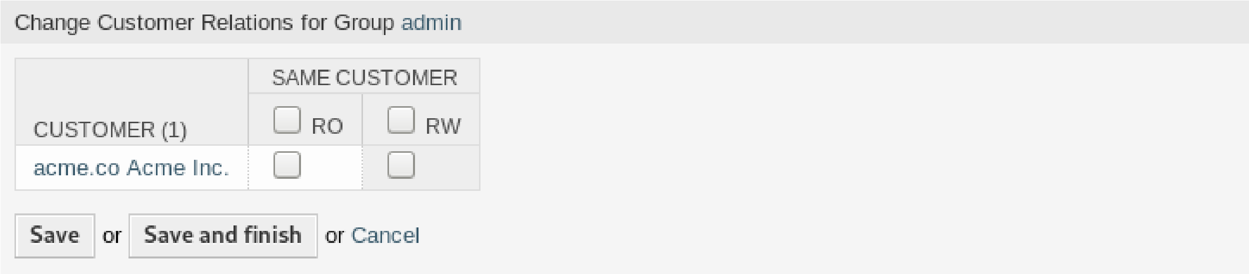
1. Click on a group in the *Groups* column.
2. Select the permissions you would like to connect the group to customers with.
3. Click on the *Save* or *Save and finish* button.



Change Group Relations for Customer Acme Inc.

GROUP	SAME CUSTOMER	
	□ RO	□ RW
admin	□	□
faq	□	□
faq_admin	□	□
faq_approval	□	□
stats	□	□

Fig. 17: Change Group Relations for Customer



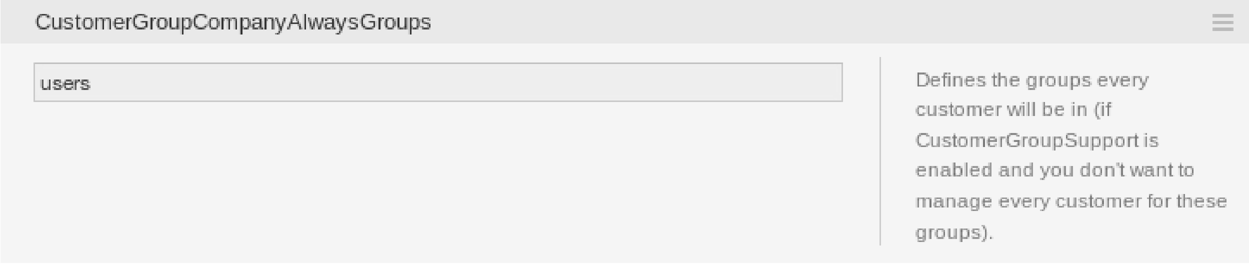
Change Customer Relations for Group admin

CUSTOMER (1)	SAME CUSTOMER	
	□ RO	□ RW
acme.co Acme Inc.	□	□

Fig. 18: Change Customer Relations for Group

To change customer default groups:

1. Click on the *Edit Customer Default Groups* button in the left sidebar.
2. Add or modify groups in setting CustomerGroupCompanyAlwaysGroups.
3. Deploy the modified system configurations.



CustomerGroupCompanyAlwaysGroups

users

Defines the groups every customer will be in (if CustomerGroupSupport is enabled and you don't want to manage every customer for these groups).

Fig. 19: `CustomerGroupCompanyAlwaysGroups` System Configuration Screen

These groups are automatically assigned to all customers.

Note: If several customers or groups are added to the system, use the search box to find a particular customer or use the filter box to find a particular group by just typing the name to filter.

Multiple customers or groups can be assigned in both screens at the same time. Additionally clicking on a customer or clicking on a group in the relations will open the *Edit Customer* screen or the *Edit Group* screen accordingly.

Warning: Accessing a customer or a group provides no back link to the relations screen.

4.5.2 Customers   Groups Relations Reference

When assigning a customer to a group or vice versa, several permissions can be set as connection between a customer and a group. The following permissions are available by default:

Same Customer Gives customer users group based access to tickets from customer users of the same customer (ticket `CustomerID` is a `CustomerID` of the customer user).

ro Read only access to the resource.

rw Full read and write access to the resource.

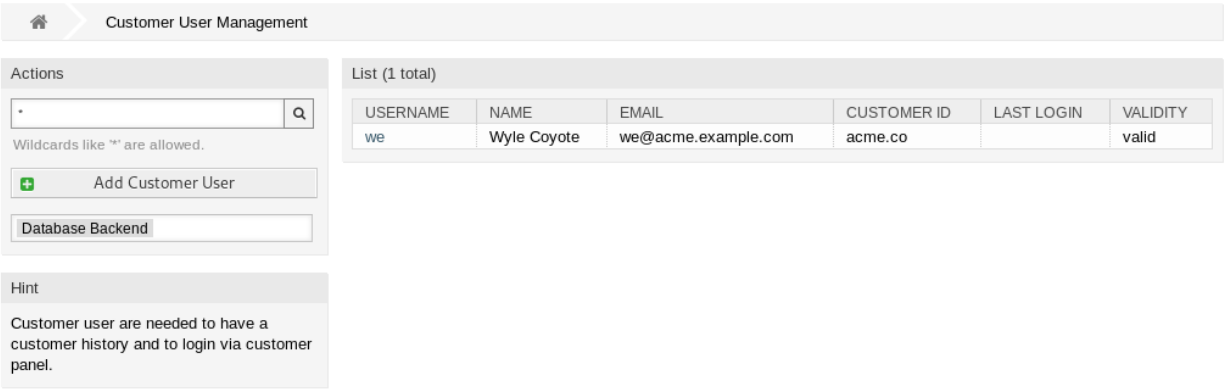
Note: By setting a checkbox in the header of a column will set all the checkboxes in the selected column.

4.6 Customer Users

A record of who your company deals with requires more information about that individual: physical location for shipping and billing purposes, as well as contact information for email and phone contact.

OTRS offers a great way to save individual information about contacts within organizations which your company serves. You can add as many personal connections into OTRS as needed.

Use this screen to add a customer user to the system. A fresh OTRS installation contains no customer users by default. The customer user management screen is available in the *Customer Users* module of the *Users, Groups & Roles* group.



Wildcards like '*' are allowed.

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

List (1 total)

USERNAME	NAME	EMAIL	CUSTOMER ID	LAST LOGIN	VALIDITY
we	Wyle Coyote	we@acme.example.com	acme.co		valid

Fig. 20: Customer User Management Screen

4.6.1 Manage Customer Users

> **Warning:** A customer user can only be added to the system when at least one customer exists. Create one or more Customers first.

> **Note:** Adding or editing a customer user is possible only by using database back end. Using external directory services like LDAP will disable the customer user management functionality.

To add a customer user:

1. Click on the *Add Customer User* button in the left sidebar.
2. Fill in the required fields.
3. Click on the *Save* button.

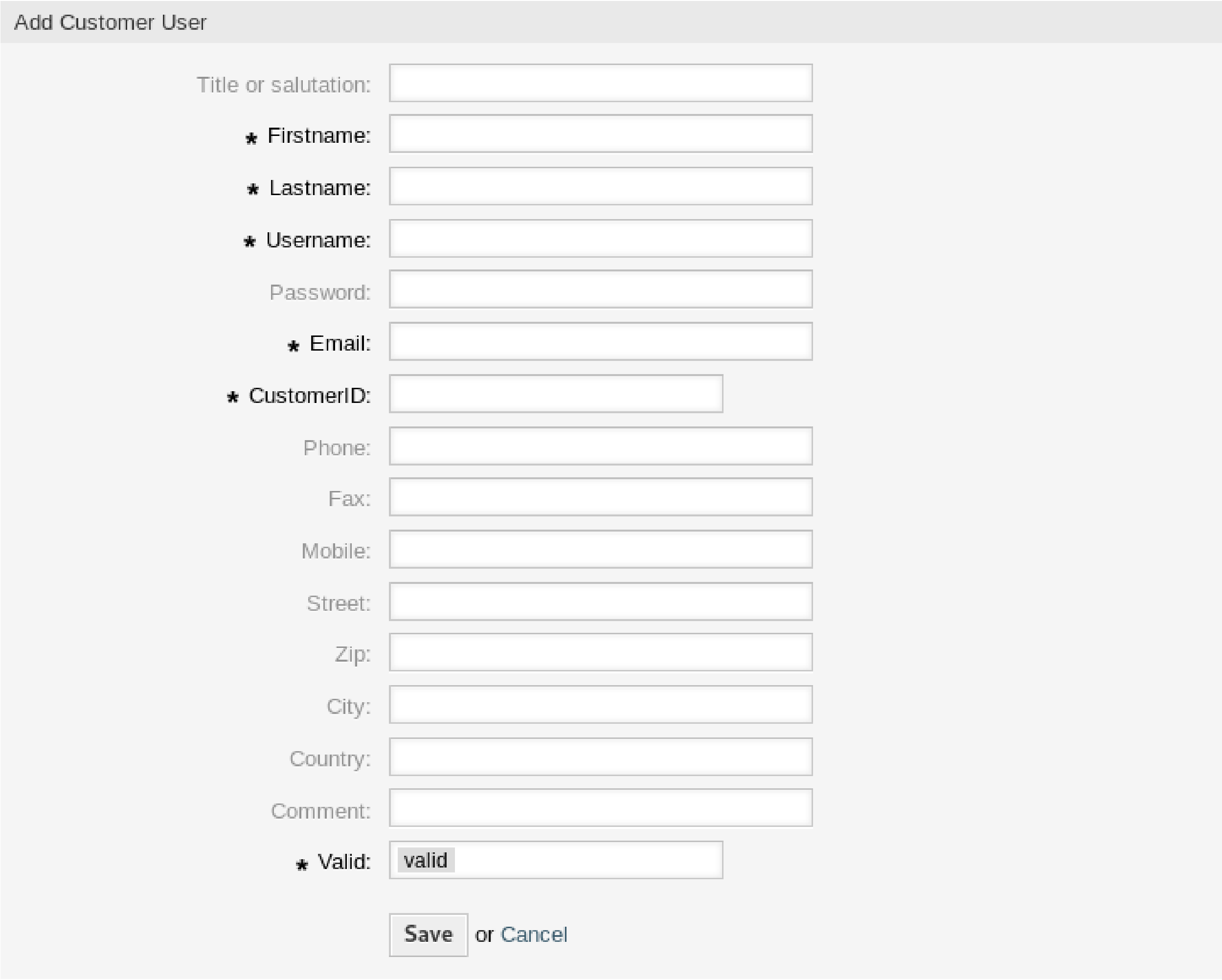
> **Warning:** Customer users can not be deleted from the system. They can only be deactivated by setting the *Validity* option to *invalid* or *invalid-temporarily*.

To edit a customer user:

1. Click on a customer user in the list of customer users.
2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.

To find a customer user:

1. Enter a search term to the search box in the left sidebar.
2. Click on the magnifying glass icon in the right part of the field or hit an `Enter`.



Add Customer User

Title or salutation:

* Firstname:

* Lastname:

* Username:

Password:

* Email:

* CustomerID:

Phone:

Fax:

Mobile:

Street:

Zip:

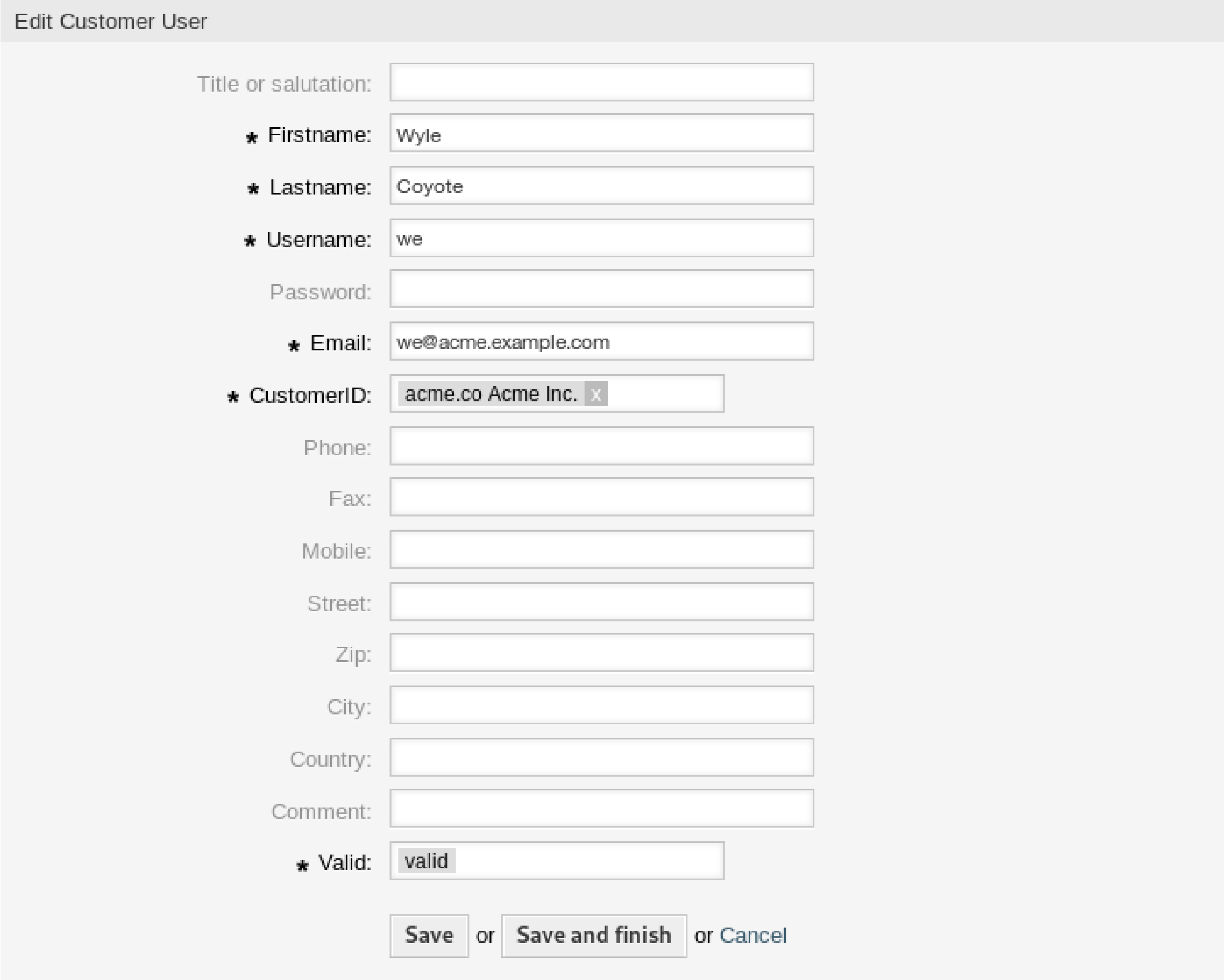
City:

Country:

Comment:

* Valid: valid

Fig. 21: Add Customer User Screen



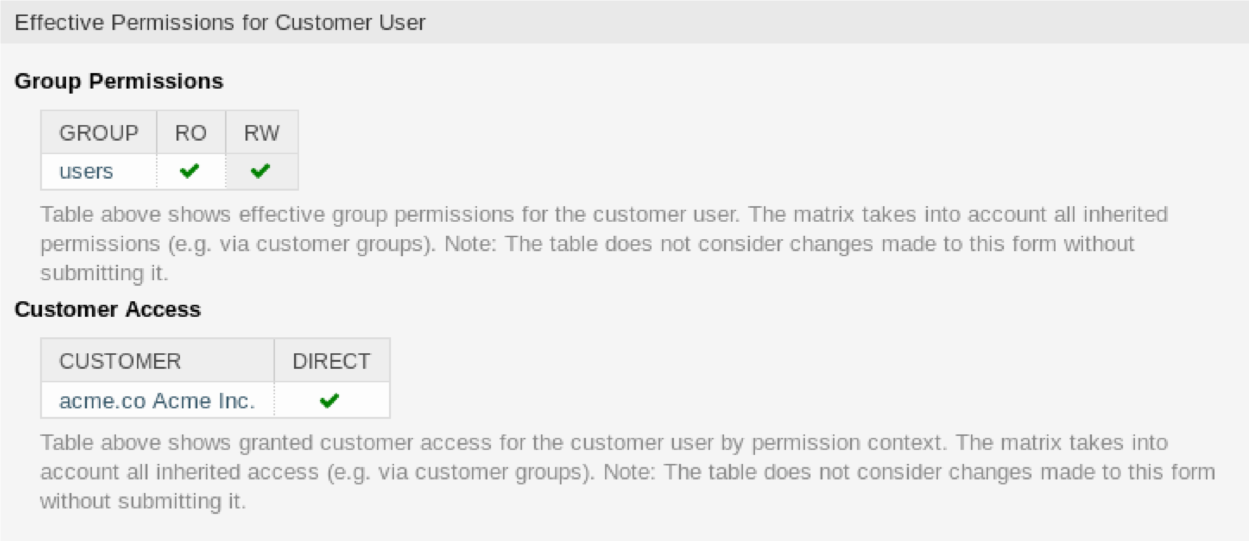
Edit Customer User

Title or salutation:	
* Firstname:	Wyle
* Lastname:	Coyote
* Username:	we
Password:	
* Email:	we@acme.example.com
* CustomerID:	acme.co Acme Inc. x
Phone:	
Fax:	
Mobile:	
Street:	
Zip:	
City:	
Country:	
Comment:	
* Valid:	valid

Fig. 22: Edit Customer User Screen

Note: If several customer users are added to the system, use the search box to find a particular customer user. Only the first 1000 customer users are listed by default.

The customer user permissions can be controlled by adding a customer or a customer user to *Groups*. This can result a complex matrix of permissions. The effective permissions for a customer user can be checked in the bottom of the *Edit Customer User* screen.



Effective Permissions for Customer User

Group Permissions

GROUP	RO	RW
users	✓	✓

Table above shows effective group permissions for the customer user. The matrix takes into account all inherited permissions (e.g. via customer groups). Note: The table does not consider changes made to this form without submitting it.

Customer Access

CUSTOMER	DIRECT
acme.co Acme Inc.	✓

Table above shows granted customer access for the customer user by permission context. The matrix takes into account all inherited access (e.g. via customer groups). Note: The table does not consider changes made to this form without submitting it.

Fig. 23: Effective Permissions for Customer User Widget

See also:

Customer Users *Groups* needs to be enabled to use this feature.

4.6.2 Customer User Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Note: These are the default fields available for the internal database table.

Title or salutation Some name prefix can be added here like *Mr.*, *Dr.*, *Jr.*, etc.

Firstname * The first name of the customer user.

Lastname * The last name of the customer user.

Username * The username of the customer user to login to the system.

Password The password of the customer user. This will be auto-generated if left empty.

Email * The email address of the customer user.

Customer * The customer company the customer user belongs to. Select a customer from the list of *Customers*.

Phone The phone number of the customer user.

Fax The fax number of the customer user.

Mobile The cellphone number of the customer user.

Street The street name of the customer.

Zip The zip code of the customer.

City The headquarter city of the customer.

Country The country of the customer user.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

Validity * Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.

See also:

It is possible to assign multiple customers to customer users via the *Customer Users*   *Customers* screen.

4.6.3 Customer User Back Ends

The system works with many customer user data attributes such as username, email address, phone number, etc. These attributes are displayed in both the agent and the external interface, and also used for the authentication of customer users.

Customer data used or displayed within the system is highly customizable. The user login and the email address are always needed for customer authentication.

You can use two types of customer back end: database or LDAP. If you already have another customer back end (e.g. SAP), it is possible to write a module that uses it.

The administrator interface does not support the configuration of external back ends. Administrators need to edit the file `Kernel/Config.pm` by copying and pasting code snippets from `Kernel/Config/Defaults.pm` manually in case of using *On-Premise* system or *((OTRS)) Community Edition*.

Warning: Do not modify the file `Kernel/Config/Defaults.pm`, it will be overwritten after upgrading the system! Copy and paste the code snippets into `Kernel/Config.pm` instead.

Note: The following configurations are taken care of by the *Customer Solution Team* in **OTRS**. Please contact the *Customer Solution Team* via support@otrs.com or in the OTRS Portal.

Database

This is the default customer user back end for new installations. The example below shows the configuration of a database customer back end, which uses customer user data stored in the database table `customer_user`.

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

(continues on next page)

(continued from previous page)

```
        Name   => Translatable('Database Backend'),
        Module => 'Kernel::System::CustomerUser::DB',
        Params => {

            # if you want to use an external database, add the
            # required settings
#            DSN  => 'DBI:odbc:yourdsn',
#            Type => 'mssql', # only for ODBC connections
#            DSN => 'DBI:mysql:database=customerdb;host=customerdbhost',
#            User => '',
#            Password => '',
            Table => 'customer_user',
#            ForeignDB => 0,    # set this to 1 if your table does not have␣
↪create_time, create_by, change_time and change_by fields

            # CaseSensitive defines if the data storage of your DBMS is case␣
↪sensitive and will be
            # preconfigured within the database driver by default.
            # If the collation of your data storage differs from the default␣
↪settings,
            # you can set the current behavior ( either 1 = CaseSensitive or␣
↪0 = CaseINSensitive )
            # to fit your environment.
            #
#            CaseSensitive => 0,

            # SearchCaseSensitive will control if the searches within the␣
↪data storage are performed
            # case sensitively (if possible) or not. Change this option to 1,␣
↪if you want to search case sensitive.
            # This can improve the performance dramatically on large␣
↪databases.
            SearchCaseSensitive => 0,
        },

        # customer unique id
        CustomerKey => 'login',

        # customer #
        CustomerID    => 'customer_id',
        CustomerValid => 'valid_id',

        # The last field must always be the email address so that a valid
        #   email address like "John Doe" <john.doe@domain.com> can be␣
↪constructed from the fields.
        CustomerUserListFields => [ 'first_name', 'last_name', 'email' ],

#        CustomerUserListFields => ['login', 'first_name', 'last_name',
↪'customer_id', 'email'],
        CustomerUserSearchFields           => [ 'login', 'first_name', 'last_
↪name', 'customer_id' ],
        CustomerUserSearchPrefix          => '*',
```

(continues on next page)

(continued from previous page)

```
        CustomerUserSearchSuffix          => '*',
        CustomerUserSearchListLimit        => 250,
        CustomerUserPostMasterSearchFields => ['email'],
        CustomerUserNameFields             => [ 'title', 'first_name', 'last_
↪name' ],
        CustomerUserEmailUniqCheck         => 1,

#         # Configures the character for joining customer user name parts. Join␣
↪single space if it is not defined.
#         # CustomerUserNameFieldsJoin => '',

#         # show now own tickets in customer panel, CompanyTickets
#         CustomerUserExcludePrimaryCustomerID => 0,
#         # generate auto logins
#         AutoLoginCreation => 0,
#         # generate auto login prefix
#         AutoLoginCreationPrefix => 'auto',
#         # admin can change customer preferences
#         AdminSetPreferences => 1,
        # use customer company support (reference to company, See␣
↪CustomerCompany settings)
        CustomerCompanySupport => 1,
        # cache time to live in sec. - cache any database queries
        CacheTTL => 60 * 60 * 24,
#         # Consider this source read only.
#         ReadOnly => 1,
        Map => [

            # Info about dynamic fields:
            #
            # Dynamic Fields of type CustomerUser can be used within the␣
↪mapping (see example below).
            # The given storage (third column) then can also be used within␣
↪the following configurations (see above):
            # CustomerUserSearchFields, CustomerUserPostMasterSearchFields,␣
↪CustomerUserListFields, CustomerUserNameFields
            #
            # Note that the columns 'frontend' and 'readonly' will be ignored␣
↪for dynamic fields.

            # note: Login, Email and CustomerID needed!
            # var, frontend, storage, shown (1=always,2=lite), required,␣
↪storage-type, http-link, readonly, http-link-target, link class(es)
            [ 'UserTitle',        Translatable('Title or salutation'), 'title
↪',           1, 0, 'var', '', 0, undef, undef ],
            [ 'UserFirstname',    Translatable('Firstname'),           'first_
↪name',      1, 1, 'var', '', 0, undef, undef ],
            [ 'UserLastname',     Translatable('Lastname'),            'last_
↪name',       1, 1, 'var', '', 0, undef, undef ],
            [ 'UserLogin',        Translatable('Username'),            'login
↪',           1, 1, 'var', '', 0, undef, undef ],
            [ 'UserPassword',     Translatable('Password'),            'pw', ␣
↪            0, 0, 'var', '', 0, undef, undef ],
```

(continues on next page)

(continued from previous page)

```
            [ 'UserEmail',         Translatable('Email'),               'email
↪',          1, 1, 'var', '', 0, undef, undef ],
#            [ 'UserEmail',         Translatable('Email'),               'email
↪',          1, 1, 'var', '[% Env("CGIHandle") %]?Action=AgentTicketCompose;
↪ResponseID=1;TicketID=[% Data.TicketID | uri %];ArticleID=[% Data.ArticleID␣
↪| uri %]', 0, '', 'AsPopup OTRSPopup_TicketAction' ],
            [ 'UserCustomerID',    Translatable('CustomerID'),
↪'customer_id',     0, 1, 'var', '', 0, undef, undef ],
#            [ 'UserCustomerIDs',   Translatable('CustomerIDs'),
↪'customer_ids',    1, 0, 'var', '', 0, undef, undef ],
            [ 'UserPhone',         Translatable('Phone'),               'phone
↪',          1, 0, 'var', '', 0, undef, undef ],
            [ 'UserFax',           Translatable('Fax'),                 'fax',␣
↪           1, 0, 'var', '', 0, undef, undef ],
            [ 'UserMobile',        Translatable('Mobile'),              'mobile
↪',         1, 0, 'var', '', 0, undef, undef ],
            [ 'UserStreet',        Translatable('Street'),              'street
↪',         1, 0, 'var', '', 0, undef, undef ],
            [ 'UserZip',           Translatable('Zip'),                 'zip',␣
↪           1, 0, 'var', '', 0, undef, undef ],
            [ 'UserCity',          Translatable('City'),                'city',
↪           1, 0, 'var', '', 0, undef, undef ],
            [ 'UserCountry',       Translatable('Country'),
↪'country',         1, 0, 'var', '', 0, undef, undef ],
            [ 'UserComment',       Translatable('Comment'),
↪'comments',        1, 0, 'var', '', 0, undef, undef ],
            [ 'ValidID',           Translatable('Valid'),               'valid_
↪id',       0, 1, 'int', '', 0, undef, undef ],

            # Dynamic field example
#            [ 'DynamicField_Name_X', undef, 'Name_X', 0, 0, 'dynamic_field',␣
↪undef, 0, undef, undef ],
        ],

        # default selections
        Selections => {

#            UserTitle => {
#                'Mr.' => Translatable('Mr.'),
#                'Mrs.' => Translatable('Mrs.'),
#            },
        },
    };
```

If you want to customize the customer user data, change the columns or add new ones to the `customer_user` table in the database.

For example, to add a new field for room number:

1. Add a new column `room` to table `customer_user`.

   MySQL or MariaDB:

```
root> mysql -u root -p -e 'ALTER TABLE otrs.customer_user ADD room␣
↪VARCHAR (250)'
```

PostgreSQL (from the `/opt/otrs` directory):

```
otrs> psql -c 'ALTER TABLE customer_user ADD COLUMN room varchar(250)'
```

2. Copy the `$Self->{CustomerUser}` section from `Kernel/Config/Defaults.pm` into `Kernel/Config.pm`.
3. Add the new column to the `Map` array.

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

You can set the HTTP link target and link class (the last two keys) to `undef` in map array elements, if they are not to be used. These keys add `target=""` and `class=""` attributes to the HTTP link element, respectively. They are ignored if HTTP link is not set (it is `''` in this example).

Note: It is recommended to always use English words for names.

See also:

Names can be translated into other languages with custom translation files. For more information, see the *Custom Translation File* chapter in the developer manual.

LDAP

If you have an LDAP directory with your customer user data, you can use it as the customer user back end. The example below shows the configuration of a LDAP customer user back end.

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

(continues on next page)

(continued from previous page)

```
#            # SourceCharset => 'iso-8859-1',
            # die if backend can't work, e. g. can't connect to server
            Die => 0,
            # Net::LDAP new params (if needed - for more info see perldoc␣
↪Net::LDAP)
            Params => {
                port    => 389,
                timeout => 120,
                async   => 0,
                version => 3,
            },
        },
        # customer unique id
        CustomerKey => 'uid',
        # customer #
        CustomerID => 'mail',
        CustomerUserListFields => ['cn', 'mail'],
        CustomerUserSearchFields => ['uid', 'cn', 'mail'],
        CustomerUserSearchPrefix => '',
        CustomerUserSearchSuffix => '*',
        CustomerUserSearchListLimit => 250,
        CustomerUserPostMasterSearchFields => ['mail'],
        CustomerUserNameFields => ['givenname', 'sn'],
        # Configures the character for joining customer user name parts. Join␣
↪single space if it is not defined.
        CustomerUserNameFieldsJoin => '',
        # show customer user and customer tickets in the external interface
        CustomerUserExcludePrimaryCustomerID => 0,
        # add a ldap filter for valid users (expert setting)
#        # CustomerUserValidFilter => '(!(description=gesperrt))',
        # admin can't change customer preferences
        AdminSetPreferences => 0,
        # cache time to live in sec. - cache any ldap queries
#        CacheTTL => 0,
        Map => [
            # note: Login, Email and CustomerID needed!
            # var, frontend, storage, shown (1=always,2=lite), required,␣
↪storage-type, http-link, readonly, http-link-target, link class(es)
            [ 'UserTitle',       Translatable('Title or salutation'), 'title',
↪               1, 0, 'var', '', 1, undef, undef ],
            [ 'UserFirstname',   Translatable('Firstname'),
↪'givenname',           1, 1, 'var', '', 1, undef, undef ],
            [ 'UserLastname',    Translatable('Lastname'),            'sn',  ␣
↪               1, 1, 'var', '', 1, undef, undef ],
            [ 'UserLogin',       Translatable('Username'),            'uid', ␣
↪               1, 1, 'var', '', 1, undef, undef ],
            [ 'UserEmail',       Translatable('Email'),               'mail',␣
↪               1, 1, 'var', '', 1, undef, undef ],
            [ 'UserCustomerID',  Translatable('CustomerID'),          'mail',␣
↪               0, 1, 'var', '', 1, undef, undef ],
            # [ 'UserCustomerIDs', Translatable('CustomerIDs'),
↪'second_customer_ids', 1, 0, 'var', '', 1, undef, undef ],
```

(continues on next page)

(continued from previous page)

```
            [ 'UserPhone',       Translatable('Phone'),
↪'telephonenumber',      1, 0, 'var', '', 1, undef, undef ],
            [ 'UserAddress',     Translatable('Address'),
↪'postaladdress',        1, 0, 'var', '', 1, undef, undef ],
            [ 'UserComment',     Translatable('Comment'),
↪'description',          1, 0, 'var', '', 1, undef, undef ],

            # this is needed, if "SMIME::FetchFromCustomer" is active
            # [ 'UserSMIMECertificate', 'SMIMECertificate', 'userSMIMECertificate
↪', 0, 1, 'var', '', 1, undef, undef ],

            # Dynamic field example
            # [ 'DynamicField_Name_X', undef, 'Name_X', 0, 0, 'dynamic_field',␣
↪undef, 0, undef, undef ],
        ],
    };
```

To activate and configure the LDAP back end:

1. Copy the `$Self->{CustomerUser}` section from `Kernel/Config/Defaults.pm` into `Kernel/Config.pm`.
2. Remove the comments (`#` characters) from the beginning of the lines.

If additional customer user attributes are stored in your LDAP directory, such as a manager name, a mobile phone number, or a department, this information can be displayed in OTRS.

To display additional customer user attributes from LDAP directory:

1. Expand the `Map` array in `Kernel/Config.pm` with the entries for these attributes.

```
[ 'UserMobilePhone', 'Mobile Phone', 'mobilephone', 1, 0, 'var', '', 1,␣
↪undef, undef ],
```

Note: It is recommended to always use English words for names.

See also:

Names can be translated into other languages with custom translation files. For more information, see the *Custom Translation File* chapter in the developer manual.

4.6.4 Multiple Customer User Back Ends

If you want to use more than one customer user data source, the `CustomerUser` configuration parameter should be expanded with a number, like `CustomerUser1` and `CustomerUser2`.

The following configuration example shows usage of both a database and an LDAP customer user back end.

```
# Data source 1: customer user database back end and settings.
    $Self->{CustomerUser1} = {
        Name   => 'Database Backend',
        Module => 'Kernel::System::CustomerUser::DB',
```

(continues on next page)

(continued from previous page)

```
        Params => {
            DSN => 'DBI:odbc:yourdsn',
            DSN => 'DBI:mysql:database=customerdb;host=customerdbhost',
            User => '',
            Password => '',
            Table => 'customer_user',
        },
        # Other setting here.
    };

# Data source 2: customer user LDAP back end and settings.
    $Self->{CustomerUser2} = {
        Name => 'LDAP Backend',
        Module => 'Kernel::System::CustomerUser::LDAP',
        Params => {
            Host => 'bay.csuhayward.edu',
            BaseDN => 'ou=seas,o=csuh',
            SSCOPE => 'sub',
            UserDN => '',
            UserPw => '',
            AlwaysFilter => '',
            Die => 0,
            Params => {
                port    => 389,
                timeout => 120,
                async   => 0,
                version => 3,
            },
        },
        # Other setting here.
    };
```

It is possible to integrate up to 10 different customer back ends. Use the *Customer Users* screen to view or edit (assuming write access is enabled) all customer user data.

4.6.5 Customer User Data in Dynamic Fields

Sometimes it can be useful to also store customer user data directly in dynamic fields of a ticket, for example to include this data in special statistics later.

The dynamic field values will be set when a ticket is created or when the customer user of a ticket is changed. The values of the dynamic fields are taken from the customer user data. This works for all back ends, but is especially useful for LDAP back ends.

To activate this optional feature:

1. Activate the setting `Ticket::EventModulePost###4100-DynamicFieldFromCustomerUser`.
2. Activate the setting `DynamicFieldFromCustomerUser::Mapping`, and modify its value. This setting should contain the map between customer user field names and names of dynamic fields which will inherit their values.
3. Create the dynamic fields, if the dynamic fields are not present in the system yet.

4. Enable display of the dynamic fields in the *Properties* widget, so you can check their current values easily. You can do this via *following instructions*.

Note: The dynamic fields in question must not be enabled in the following action forms:

- `Forms###AgentFrontend::TicketCreate::Email::CreateProperties`
- `Forms###AgentFrontend::TicketCreate::Phone::CreateProperties`
- `Forms###AgentFrontend::TicketCreate::SMS::CreateProperties`
- `Forms###AgentFrontend::Ticket::Action::Customer`

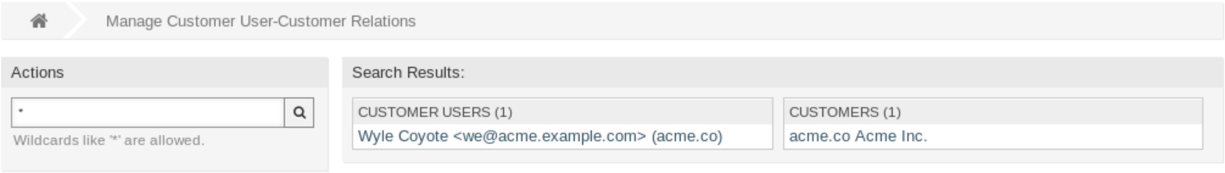
If they were, the field values from the screen would have precedence over the automatically set values.

4.7 Customer Users   Customers

In an organization, for example, which views its customers as it’ s departments and teams, a customer user may have to have access to multiple customers tickets for controlling purposes. Maybe you have partners who represent several different companies, or a corporation wants to have a look at all the requests of their subsidiaries.

For all situations, OTRS provides the means. Aside from a primary customer, your customer users can gain access to multiple customer tickets as defined by you.

Use this screen to add one or more customer users to one or more customers. To use this function, at least one customer user and one customer need to have been added to the system. The management screen is available in the *Customer Users   Customers* module of the *Users, Groups & Roles* group.


Manage Customer User-Customer Relations
Actions
*
Wildcards like '*' are allowed.
Search Results:
CUSTOMER USERS (1)
CUSTOMERS (1)

Wyle Coyote <we@acme.example.com> (acme.co)

acme.co Acme Inc.

Fig. 24: Manage Customer User-Customer Relations

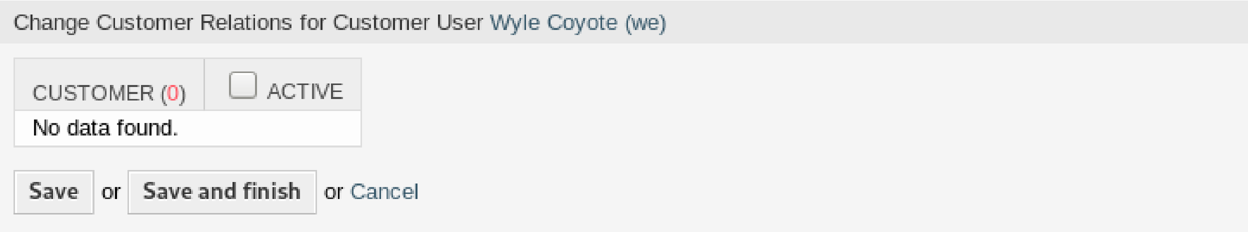
4.7.1 Manage Customer Users   Customers Relations

Note: This module is for assigning a *customer user* to additional *customer* records. The primary customer is assigned via the *Customer User Settings*.

To assign some customers to a customer user:

1. Click on a customer user in the *Customer Users* column.
2. Select the customers you would like to assign to the customer user.
3. Click on the *Save* or *Save and finish* button.

To assign some customer users to a customer:

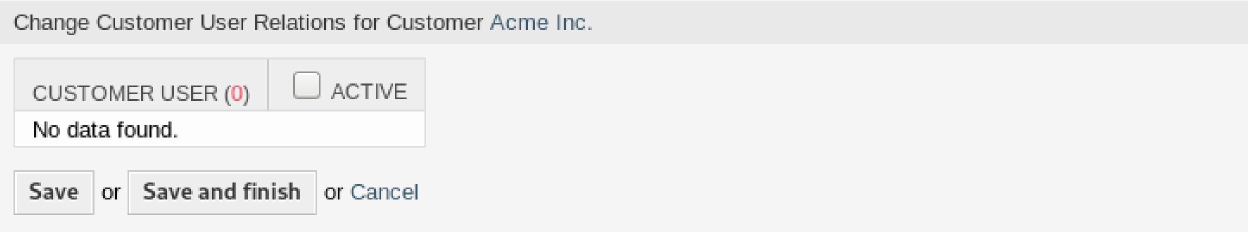


Change Customer Relations for Customer User Wyle Coyote (we)

CUSTOMER (0)	☐ ACTIVE
No data found.	

Fig. 25: Change Customer Relations for Customer User

1. Click on a customer in the *Customers* column.
2. Select the customer users you would like to assign to the customer.
3. Click on the *Save* or *Save and finish* button.



Change Customer User Relations for Customer Acme Inc.

CUSTOMER USER (0)	☐ ACTIVE
No data found.	

Fig. 26: Change Customer User Relations for Customer

In the previous versions of **OTRS** it was only possible to enable or disable access to company tickets for customer users for every customer user back end via `CustomerUserExcludePrimaryCustomerID` parameter. It was not possible to select the primary customer ID, because it was assumed to be always there anyways, and for the same reason it is still not possible if `CustomerUserExcludePrimaryCustomerID` is disabled for that customer user back end.

In order to allow privilege separation for customer users of a common company, it should be possible to remove access to tickets of the same company and then individually reassign access to specific customer users. Then these customer users have company ticket access while all others do not.

To allow privilege separation for the customer users of the same company:

1. Go to the *System Configuration* screen.
2. Search for the setting `CustomerDisableCompanyTicketAccess` and enable it to make sure not all customer users get access to company tickets until the configuration is finished.
3. Copy the `$Self->{CustomerUser}` section from `Kernel/Config/Defaults.pm` into `Kernel/Config.pm`.
4. Remove the comment (# character) from the beginning of the line contained `CustomerUserExcludePrimaryCustomerID` and set the value to `1`.

   ```
   CustomerUserExcludePrimaryCustomerID => 1,
   ```

5. Set *Customer Users* *Customers* relations for customer users who need to have access to company tickets.
6. Go to the *System Configuration* screen.
7. Search for the setting `CustomerDisableCompanyTicketAccess` and disable it to allow access to company tickets only for customer users configured in step 5.

Note: If several customer users or customers are added to the system, use the search box to find a particular customer user or customer. Only the first 1000 customer users and customers are listed by default.

Multiple customer users or customers can be assigned in both screens at the same time. Additionally clicking on a customer user or clicking on a customer in the relations screen will open the *Edit Customer User* screen or the *Edit Customer* screen accordingly.

Warning: Accessing a customer user or a customer provides no back link to the relations screen.

Note: By setting a checkbox in the header of a column will set all the checkboxes in the selected column.

4.8 Customer Users Groups

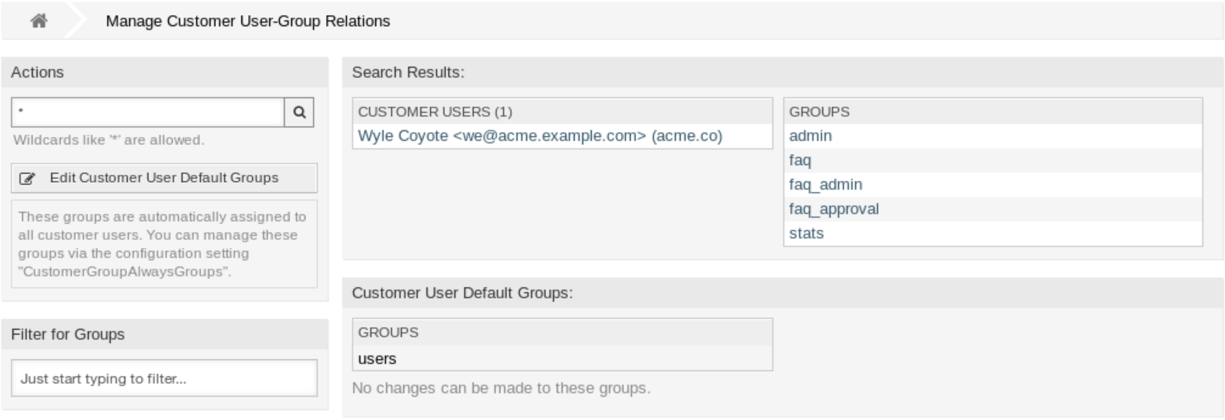
Customer users shouldn't need to be bothered with the internal workings of your service desk. A single point of contact request can trigger several processes within your organization, all of which having the customer user information attached and are visible to the customer.

OTRS allows you to assign *group* permissions to customer users. Access works just the same as for agents, preventing a customer from modifying and viewing a request. Thus allowing the customer to focus on the results of the original communication and funneling the discussion through one ticket.

See also:

Assign a group to an entire customer using *Customers Groups*.

Use this screen to add one or more customer users to one or more groups. To use this function, at least one customer user and one group need to have been added to the system. The management screen is available in the *Customers Users Groups* module of the *Users, Groups & Roles* group.



Manage Customer User-Group Relations

Actions

Wildcards like '*' are allowed.

Edit Customer User Default Groups

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

Filter for Groups

Just start typing to filter...

Search Results:

CUSTOMER USERS (1)
Wyle Coyote <we@acme.example.com> (acme.co)

GROUPS
admin
faq
faq_admin
faq_approval
stats

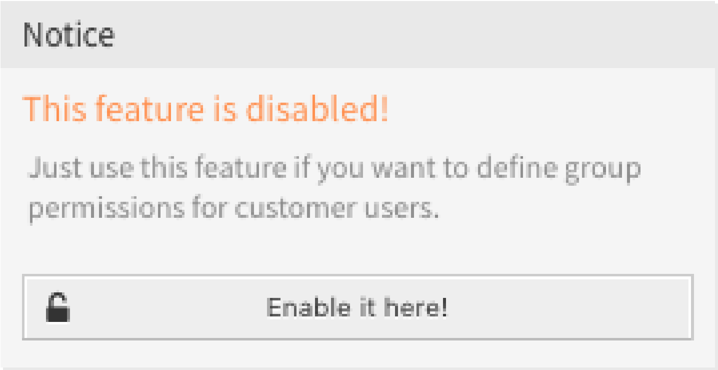
Customer User Default Groups:

GROUPS
users

No changes can be made to these groups.

Fig. 27: Manage Customer User-Group Relations

Customer group support needs to be enabled in at least one customer user *back end* to use this function. For the default OTRS *back end*, this can be enabled in the system configuration by clicking on the *Enable it here!* button.



Notice

This feature is disabled!

Just use this feature if you want to define group permissions for customer users.

Fig. 28: Enable Customer Group Feature

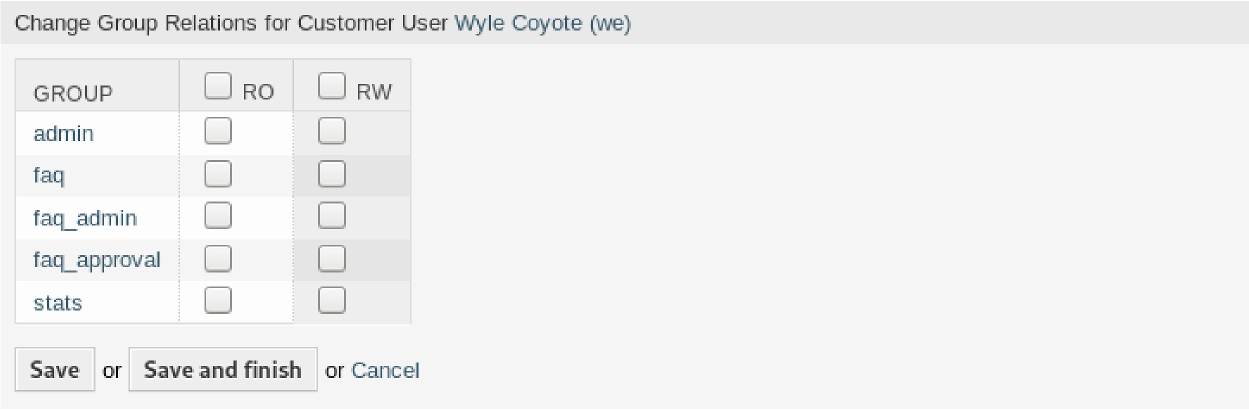
Note: To enable this feature in systems using a directory server or multiple non-default back ends, a custom configuration file needs to be placed in `Kernel/Config/Files` (for example named `ZZZ_CustomerBackend.pm`). Once activated, all customer users from this back end will require group assignment.

Warning: After making changes to the back end, the server cache will be deleted, which may cause a temporary drop in performance.

4.8.1 Manage Customer Users Groups Relations

To assign some groups to a customer user:

1. Click on a customer user in the *Customer Users* column.
2. Select the permissions you would like to connect the customer user to groups with.
3. Click on the *Save* or *Save and finish* button.



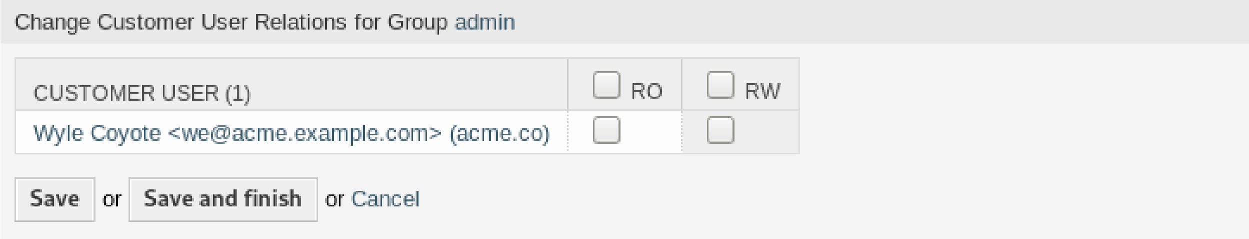
Change Group Relations for Customer User Wyle Coyote (we)

GROUP	□ RO	□ RW
admin	□	□
faq	□	□
faq_admin	□	□
faq_approval	□	□
stats	□	□

Fig. 29: Change Group Relations for Customer User

To assign some customer users to a group:

1. Click on a group in the *Groups* column.
2. Select the permissions you would like to connect the group to customer users with.
3. Click on the *Save* or *Save and finish* button.



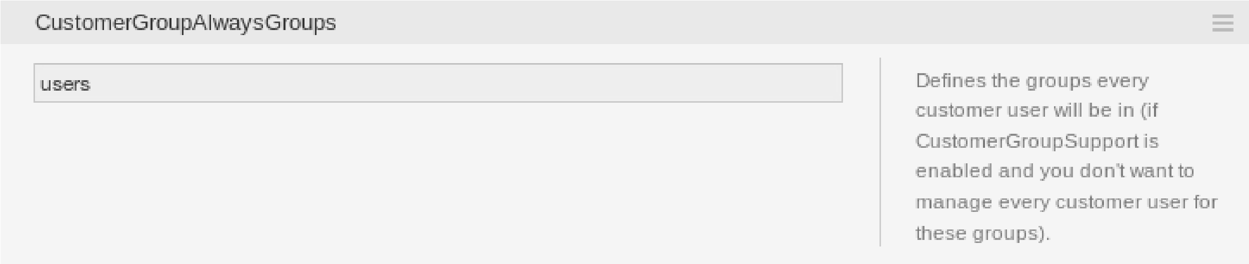
Change Customer User Relations for Group admin

CUSTOMER USER (1)	□ RO	□ RW
Wyle Coyote <we@acme.example.com> (acme.co)	□	□

Fig. 30: Change Customer User Relations for Group

To change customer user default groups:

1. Click on the *Edit Customer User Default Groups* button in the left sidebar.
2. Add or modify groups in setting CustomerGroupAlwaysGroups.
3. Deploy the modified system configurations.



CustomerGroupAlwaysGroups

users

Defines the groups every customer user will be in (if CustomerGroupSupport is enabled and you don't want to manage every customer user for these groups).

Fig. 31: `CustomerGroupAlwaysGroups` System Configuration Screen

These groups are automatically assigned to all customer users.

Note: If several customer users or groups are added to the system, use the search box to find a particular customer user or use the filter box to find a particular group by just typing the name to filter.

Multiple customer users or groups can be assigned in both screens at the same time. Additionally clicking on a customer user or clicking on a group in the relations screen will open the *Edit Customer User* screen or the *Edit Group* screen accordingly.

Warning: Accessing a customer user or a group provides no back link to the relations screen.

4.8.2 Customer Users Groups Relations Reference

When assigning a customer user to a group or vice versa, several permissions can be set as connection between a customer user and a group. The following permissions are available by default:

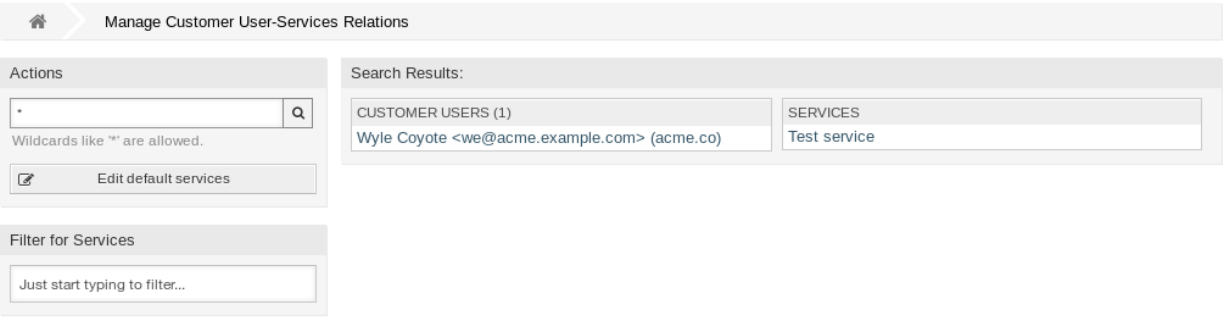
ro Read only access to the resource.

rw Full read and write access to the resource.

Note: By setting a checkbox in the header of a column will set all the checkboxes in the selected column.

4.9 Customer Users   Services

Use this screen to add one or more customer users to one or more services. To use this function, at least one customer user and one service need to have been added to the system. The management screen is available in the *Customers Users   Services* module of the *Users, Groups & Roles* group.



Manage Customer User-Services Relations

Actions

*

Wildcards like '*' are allowed.

Filter for Services

Just start typing to filter...

Search Results:

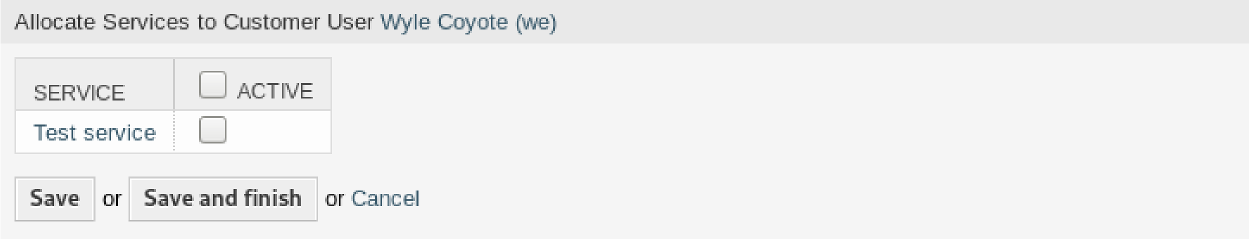
CUSTOMER USERS (1)	SERVICES
Wyle Coyote <we@acme.example.com> (acme.co)	Test service

Fig. 32: Manage Customer User-Service Relations

4.9.1 Manage Customers Users   Services Relations

To allocate some services to a customer user:

1. Click on a customer user in the *Customer Users* column.
2. Select the services you would like to allocate to the customer user.
3. Click on the *Save* or *Save and finish* button.



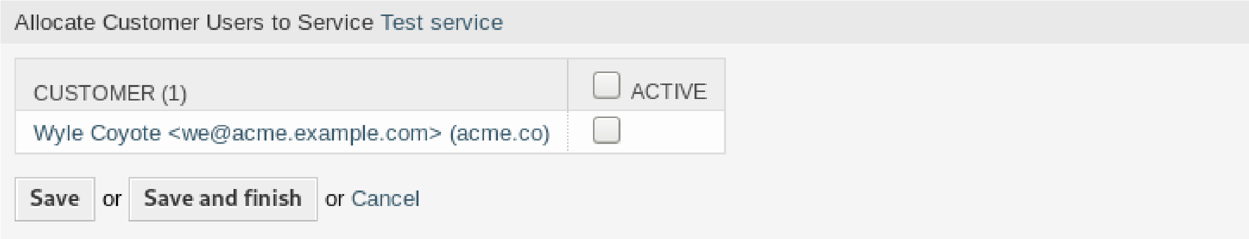
Allocate Services to Customer User Wyle Coyote (we)

SERVICE	□ ACTIVE
Test service	□

Fig. 33: Change Service Relations for Customer User

To allocate some customer users to a service:

1. Click on a service in the *Services* column.
2. Select the customer users you would like to allocate to the service.
3. Click on the *Save* or *Save and finish* button.



Allocate Customer Users to Service Test service

CUSTOMER (1)	□ ACTIVE
Wyle Coyote <we@acme.example.com> (acme.co)	□

Fig. 34: Change Customer User Relations for Service

Note: If several customer users or services are added to the system, use the search box to find a particular customer user or use the filter box to find a particular service by just typing the name to filter.

Multiple customer users or services can be assigned in both screens at the same time. Additionally clicking on a customer user or clicking on a service in the relations screen will open the *Edit Customer User* screen or the *Edit Service* screen accordingly.

Warning: Accessing a customer user or a service provides no back link to the relations screen.

Note: By setting a checkbox in the header of a column will set all the checkboxes in the selected column.

4.9.2 Manage Default Services

It is possible to add default services, so that all customer users may access them. This prevents having to add each service to each customer user.

To edit the default services:

1. Click on the *Edit default services* button in the left sidebar.
2. Select the services which should be selectable for all customer users.
3. Click on the *Save* or *Save and finish* button.

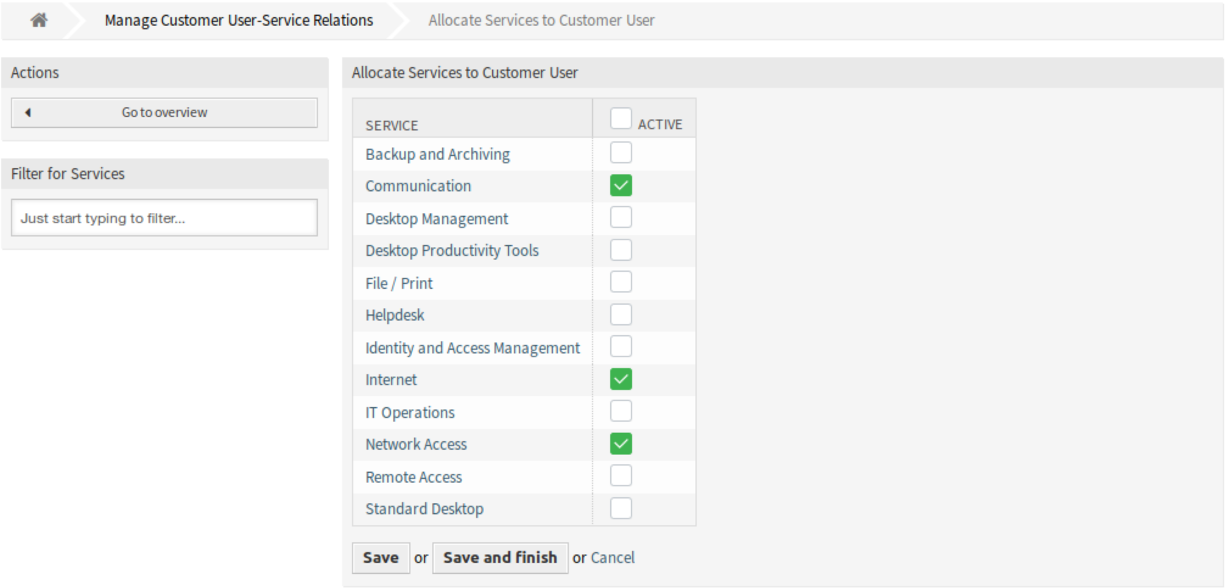
Warning: Mixing default services and customer specific services can be confusing. If a customer user has specific services assigned, then the default services will be not applied.

4.10 Groups

Use this screen to add groups to the system. A fresh OTRS installation contains some default groups. The group management screen is available in the *Groups* module of the *Users, Groups & Roles* group.

4.10.1 Manage Groups

To add a group:

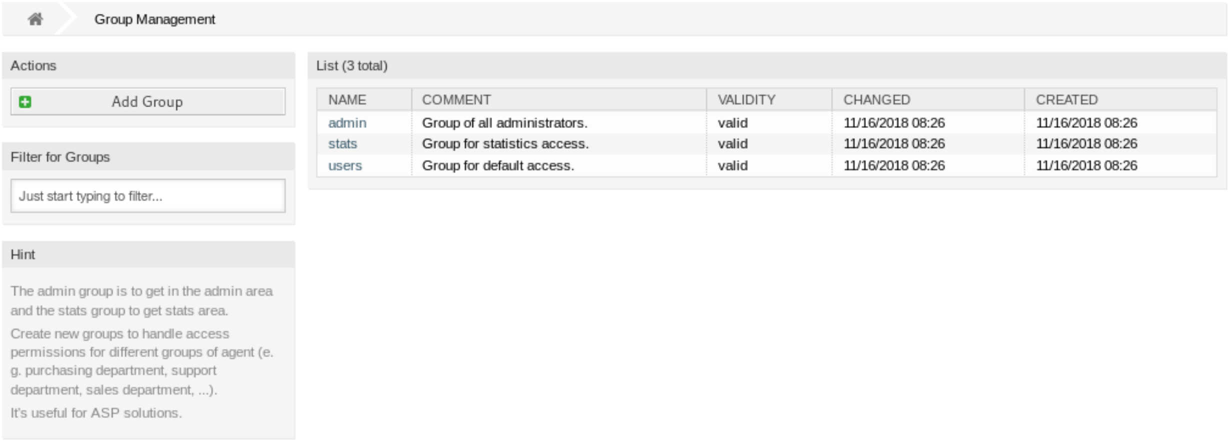


Manage Customer User-Service Relations
Allocate Services to Customer User
Actions
Go to overview
Filter for Services
Just start typing to filter...
Allocate Services to Customer User

SERVICE	□ ACTIVE
Backup and Archiving	□
Communication	☑
Desktop Management	□
Desktop Productivity Tools	□
File / Print	□
Helpdesk	□
Identity and Access Management	□
Internet	☑
IT Operations	□
Network Access	☑
Remote Access	□
Standard Desktop	□

Save or Save and finish or Cancel

Fig. 35: Allocate Services to Customer User Screen



NAME	COMMENT	VALIDITY	CHANGED	CREATED
admin	Group of all administrators.	valid	11/16/2018 08:26	11/16/2018 08:26
stats	Group for statistics access.	valid	11/16/2018 08:26	11/16/2018 08:26
users	Group for default access.	valid	11/16/2018 08:26	11/16/2018 08:26

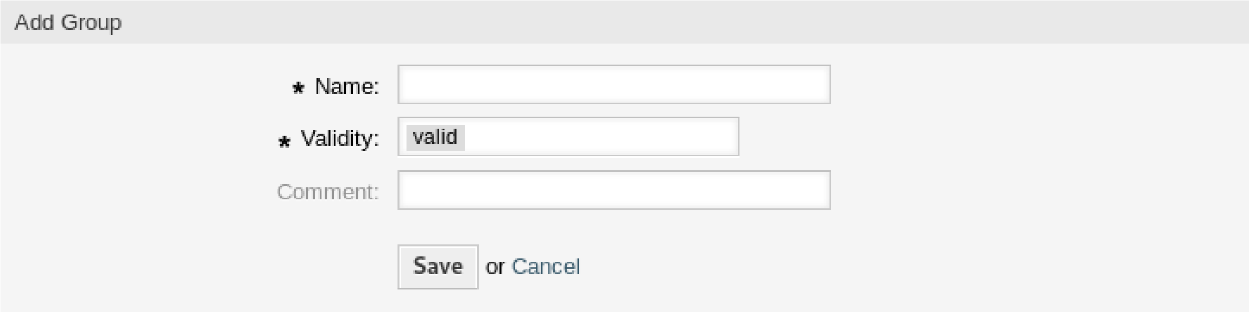
The admin group is to get in the admin area and the stats group to get stats area.

Create new groups to handle access permissions for different groups of agent (e. g. purchasing department, support department, sales department, ...).

It's useful for ASP solutions.

Fig. 36: Group Management Screen

1. Click on the *Add Group* button in the left sidebar.
2. Fill in the required fields.
3. Click on the *Save* button.



Add Group

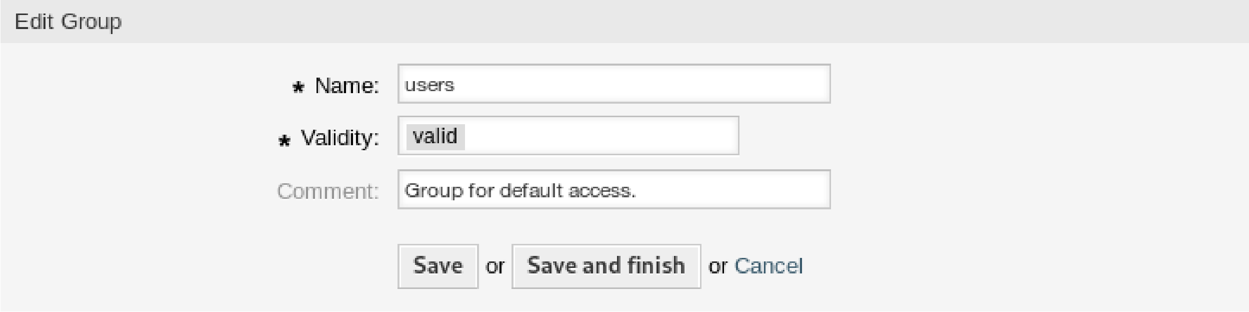
∗ Name:	
∗ Validity:	valid
Comment:	

Fig. 37: Add Group Screen

Warning: Groups can not be deleted from the system. They can only be deactivated by setting the *Validity* option to *invalid* or *invalid-temporarily*.

To edit a group:

1. Click on a group in the list of groups.
2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.



Edit Group

∗ Name:	users
∗ Validity:	valid
Comment:	Group for default access.

Fig. 38: Edit Group Screen

Note: If several groups are added to the system, use the filter box to find a particular group by just typing the name to filter.

4.10.2 Group Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

> **Note:** Renaming a group does not affect permissions previously given. When *group1* is now called *group2*, then all the permissions are the same for the users which used to be assigned to *group1*. This result is because OTRS uses IDs for the relationship, and not the name.

Validity * Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.

> **Note:** Invalidating a group does not remove the permissions from the user, but only makes them invalid. If you reactivate this group, even with a new name, the permissions take effect.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

4.10.3 Default Groups

Every agent’ s account should belong to at least one group or role. In a fresh installation, there are some pre-defined groups available:

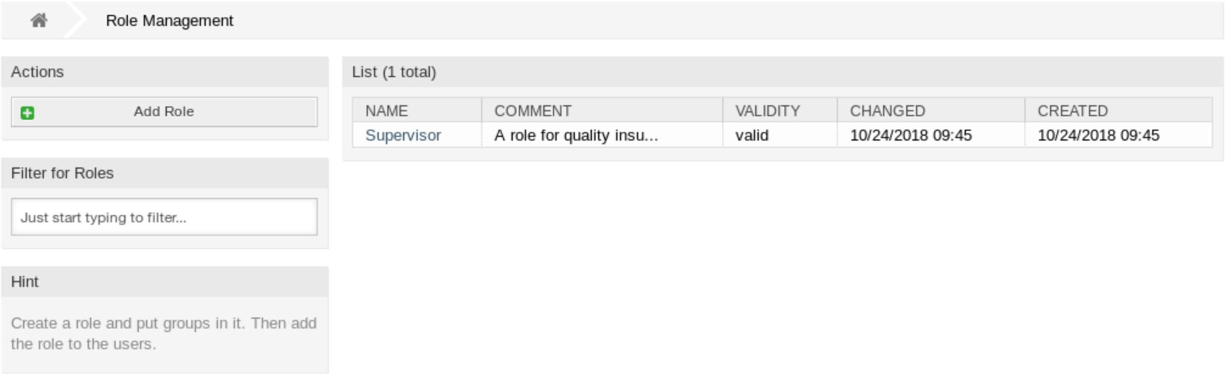
`admin` Allowed to perform administrative tasks in the system.

`stats` Qualified to access the stats module of OTRS and generate statistics.

`users` Agents should belong to this group, with read and write permissions. They can then access all functions of the ticket system.

4.11 Roles

Use this screen to add roles to the system. A fresh OTRS installation contains no roles by default. The role management screen is available in the *Roles* module of the *Users, Groups & Roles* group.



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

List (1 total)

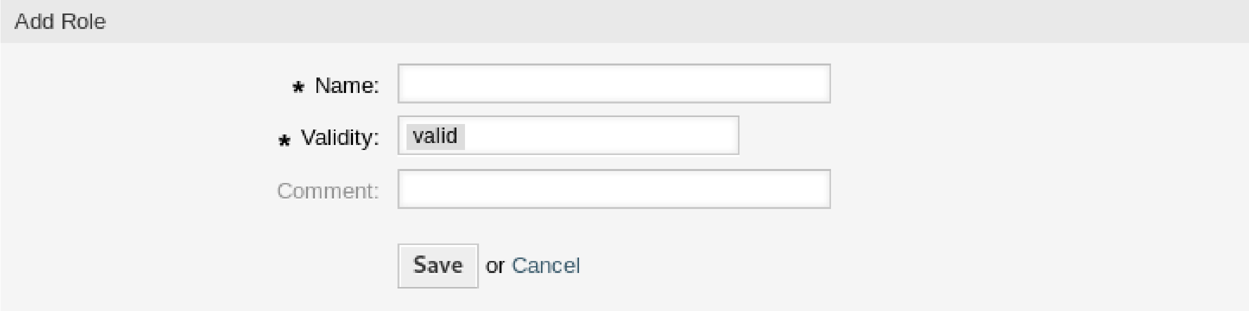
NAME	COMMENT	VALIDITY	CHANGED	CREATED
Supervisor	A role for quality insu...	valid	10/24/2018 09:45	10/24/2018 09:45

Fig. 39: Role Management Screen

4.11.1 Manage Roles

To add a role:

1. Click on the *Add Role* button in the left sidebar.
2. Fill in the required fields.
3. Click on the *Save* button.



Add Role

* Name:

* Validity: valid

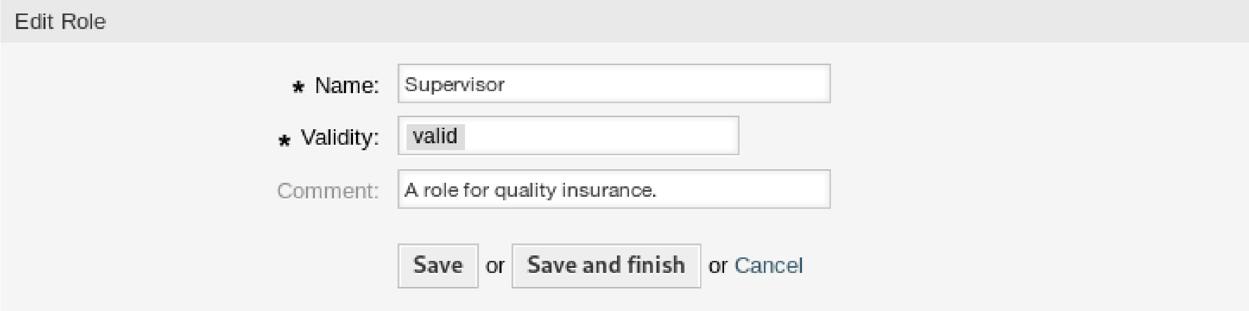
Comment:

Fig. 40: Add Role Screen

Warning: Roles can not be deleted from the system. They can only be deactivated by setting the *Validity* option to *invalid* or *invalid-temporarily*.

To edit a role:

1. Click on a role in the list of roles.
2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.



Edit Role

* Name: Supervisor

* Validity: valid

Comment: A role for quality insurance.

Fig. 41: Edit Role Screen

Note: If several roles are added to the system, use the filter box to find a particular role by just typing the name to filter.

4.11.2 Role Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

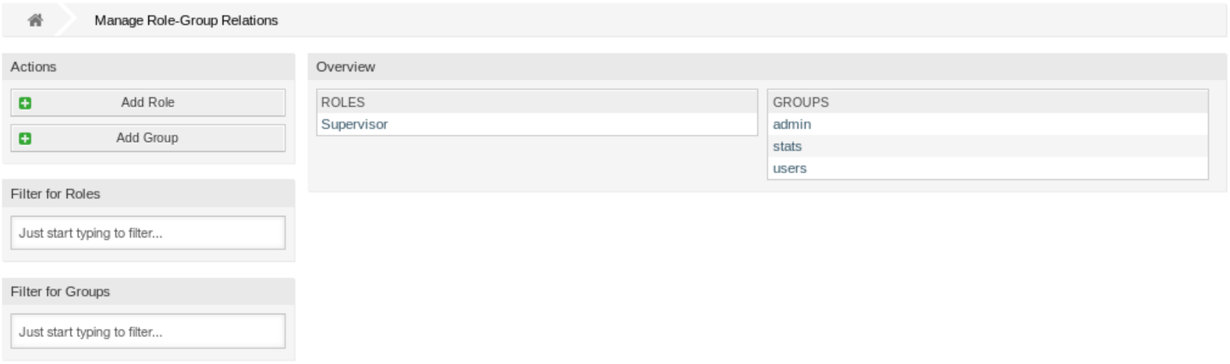
Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Validity * Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

4.12 Roles   Groups

Use this screen to add one or more roles to one or more groups. To use this function, at least one role and one group need to have been added to the system. The management screen is available in the *Roles Groups* module of the *Users, Groups & Roles* group.


Manage Role-Group Relations
Actions
Add Role
Add Group
Filter for Roles
Just start typing to filter...
Filter for Groups
Just start typing to filter...
Overview

ROLES	GROUPS
Supervisor	admin
	stats
	users

Fig. 42: Manage Role-Group Relations

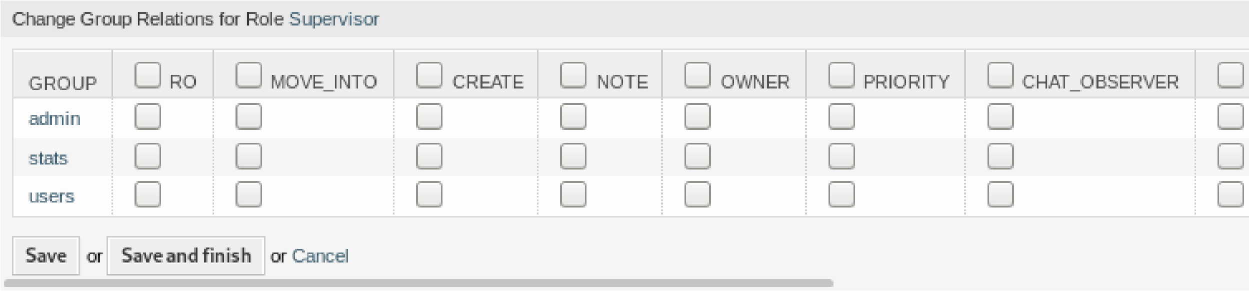
4.12.1 Manage Roles   Groups Relations

To assign some groups to a role:

1. Click on a role in the *Roles* column.
2. Select the permissions you would like to connect the role to groups with.
3. Click on the *Save* or *Save and finish* button.

To assign some roles to a group:

1. Click on a group in the *Groups* column.
2. Select the permissions you would like to connect the group to roles with.
3. Click on the *Save* or *Save and finish* button.

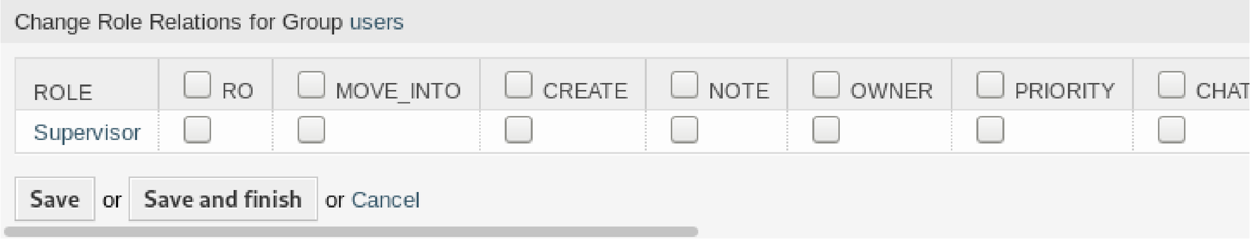


Change Group Relations for Role Supervisor

GROUP	□ RO	□ MOVE_INTO	□ CREATE	□ NOTE	□ OWNER	□ PRIORITY	□ CHAT_OBSERVER	□
admin	□	□	□	□	□	□	□	□
stats	□	□	□	□	□	□	□	□
users	□	□	□	□	□	□	□	□

Save or Save and finish or Cancel

Fig. 43: Change Group Relations for Role



Change Role Relations for Group users

ROLE	□ RO	□ MOVE_INTO	□ CREATE	□ NOTE	□ OWNER	□ PRIORITY	□ CHAT
Supervisor	□	□	□	□	□	□	□

Save or Save and finish or Cancel

Fig. 44: Change Role Relations for Group

Note: If several roles or groups are added to the system, use the filter box to find a particular role or group by just typing the name to filter.

Multiple roles or groups can be assigned in both screens at the same time. Additionally clicking on a role or clicking on a group in the relations screen will open the *Edit Role* screen or the *Edit Group* screen accordingly.

Warning: Accessing a role or a group provides no back link to the relations screen.

4.12.2 Roles   Groups Relations Reference

When assigning a role to a group or vice versa, several permissions can be set as connection between a role and a group. The following permissions are available by default:

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

move_into Permissions to move tickets into this group/queue.

create Permissions to create tickets in this group/queue.

note Permissions to add notes to tickets in this group/queue.

owner Permissions to change the owner of tickets in this group/queue.

priority Permissions to change the ticket priority in this group/queue.

chat_observer Users with this permission type will only be able to observe chats in a channel after they have been invited.

chat_participant Users with this permission type will be able to take part in a chat, but only after they get invited to it.

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.

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

See also:

Not all available permissions are shown by default. See System::Permission setting for permissions that can be added. These additional permissions can be added:

stats Gives access to the stats page.

bounce Permissions to bounce an email message (with bounce button in *Ticket Zoom* screen).

compose Permissions to compose an answer for a ticket.

customer Permissions to change the customer of a ticket.

forward Permissions to forward a message (with the forward button).

pending Permissions to set a ticket to pending.

phone Permissions to add a phone call to a ticket.

responsible Permissions to change the responsible agent for a ticket.

Note: By setting a checkbox in the header of a column will set all the checkboxes in the selected column. By setting the checkbox in the last *rw* column will set all the checkboxes in the selected row.

CHAPTER 5

Processes & Automation

Working effectively with tickets requires more than just the possibility to manually change their state, add information, communicate with other persons and finally close the tickets.

Automation frees agents from reoccurring and time-consuming tasks and allows them to focus on activities, where their interaction is required.

Process management guides customer users and agents through ticket creation to closure ensuring that tickets take defined workflows at any time.

OTRS offers many options to automate tasks based on events, time, external systems and defined processes. OTRS also includes the possibility to add individual information types to tickets and help agents to lower their error rate when working with tickets by allowing only defined activities for tickets in specific states.

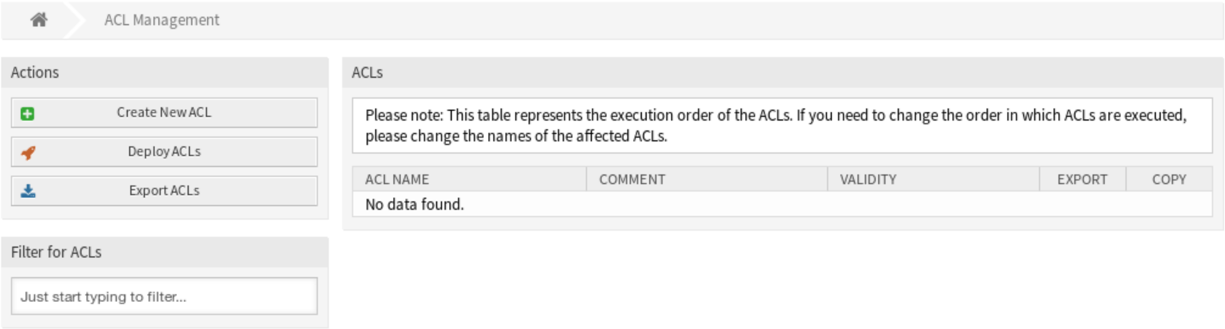
5.1 Access Control Lists (ACL)

Working with tickets can become a bewildering task. Many options are given to process, or close tickets, even if they are not needed in the current state of a ticket or due to the role of the current agent. Hiding unneeded entries cleans up the menu bar and gets it easier to work with, hiding values from dynamic fields or next queues lowers chance of human error.

OTRS uses access control lists (ACL) to restrict agents and customer users on ticket options, allowing only correct and meaningful activities with a ticket. OTRS administrators can easily generate ACLs in the graphical interface to prevent ticket closure until meeting specific requirements, prevent tickets from being moved to queues before adding the defined information and much more.

Use this screen to manage access control lists in the system. A fresh OTRS installation contains no access control lists by default. The access control lists management screen is available in the *Access Control Lists (ACL)* module of the *Processes & Automation* group.

5.1.1 Manage Access Control Lists



Please note: This table represents the execution order of the ACLs. If you need to change the order in which ACLs are executed, please change the names of the affected ACLs.

ACL NAME	COMMENT	VALIDITY	EXPORT	COPY
No data found.				

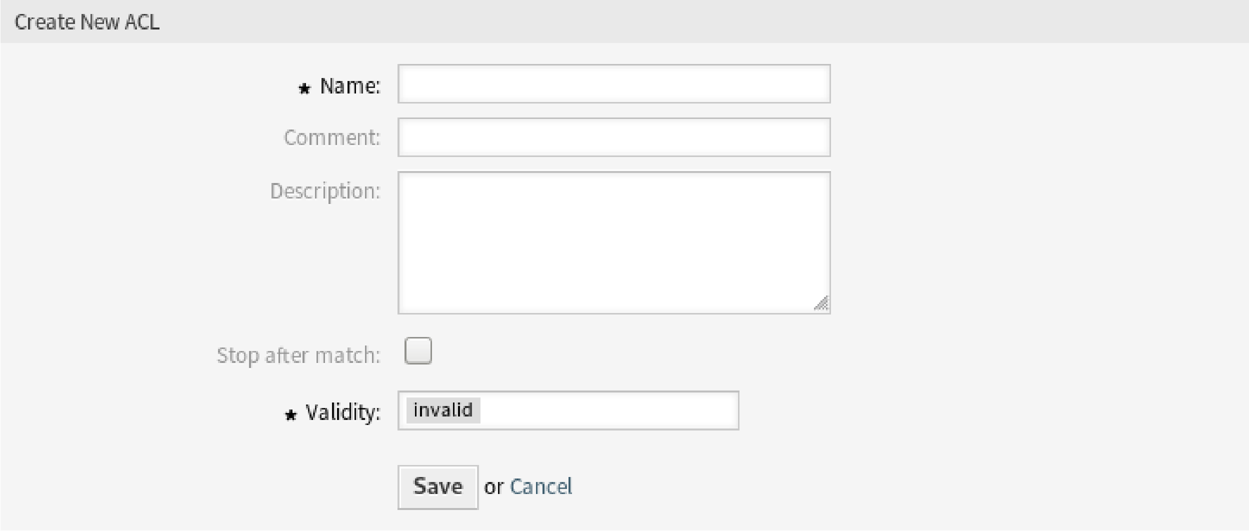
Fig. 1: ACL Management Screen

Note: When creating some access control lists, please keep in mind that they are executed alphabetically as displayed in the access control lists overview.

Warning: ACL restrictions will be ignored for the superuser account (UserID 1).

To create a new ACL:

1. Click on the *Create New ACL* button in the left sidebar.
2. Fill in the required fields.
3. Click on the *Save* button.
4. You will be redirected to *Edit ACL* screen to edit the ACL structure.



Validity: invalid

Fig. 2: Create New ACL Screen

To edit an ACL:

1. Click on an ACL in the list of ACLs or you are already redirected here from *Create New ACL* screen.

2. Modify the fields and the ACL structure.
3. Click on the *Save* or *Save and finish* button.
4. Deploy all ACLs.

To delete an ACL:

1. Click on an ACL in the list of ACLs.
2. Set the *Validity* option to *invalid* or *invalid-temporarily*.
3. Click on the *Save* button. A new *Delete Invalid ACL* button will appear in the left sidebar.
4. Click on the *Delete Invalid ACL* button.
5. Click on the *Delete* button in the confirmation screen.
6. Deploy all ACLs.

Warning: ACLs are written into `ZZZACL.pm` file in Perl format. Without deploying, all ACLs are still in this cache file even if they are deleted or the *Validity* option is set to *invalid* or *invalid-temporarily*. Don't forget to deploy all ACLs after modifications!

To deploy all ACLs:

1. Click on the *Deploy ACLs* button in the left sidebar.

Note: New or modified ACLs have to deploy in order to make affect the behavior of the system. Setting the *Validity* option to *valid* just indicates, which ACLs should be deployed.

To export all ACLs:

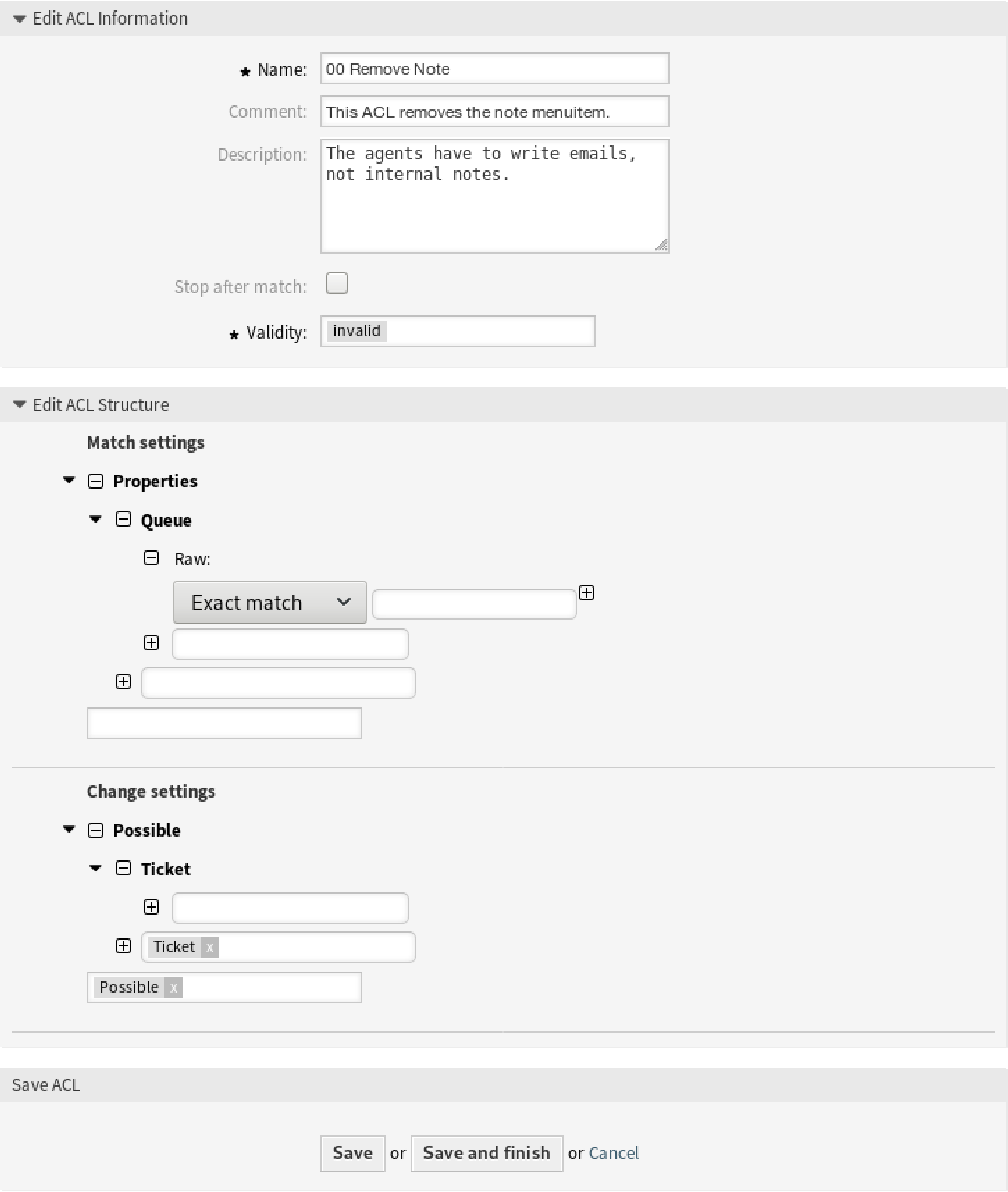
1. Click on the *Export ACLs* button in the left sidebar.
2. Choose a location in your computer to save the `Export_ACL.yml` file.

To import ACLs:

1. Click on the *Browse*… button in the left sidebar.
2. Select a previously exported `.yml` file.
3. Click on the *Overwrite existing ACLs?* checkbox, if you would like to overwrite the existing ACLs.
4. Click on the *Import ACL configuration(s)* button.
5. Deploy the imported ACLs with *Deploy ACLs* button.

Note: If several ACLs are added to the system, use the filter box to find a particular ACL by just typing the name to filter.

Warning: Changing the name of this object should be done with care, the check only provides verification for certain settings and ignores things where the name can't be verified. Some examples are dashboard filters, access control lists (ACLs), and processes (sequence flow actions) to name a few. Documentation of your setup is key to surviving a name change.



★ Name: 00 Remove Note

Comment: This ACL removes the note menuitem.

Description: The agents have to write emails, not internal notes.

Stop after match: ☐

★ Validity: invalid

Fig. 3: Edit ACL Structure Screen

5.1.2 ACL Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

Description Like comment, but longer text can be added here.

Stop after match ACLs are evaluated in alphabetical order. This setting disables the evaluation of the subsequent ACLs.

Validity * Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.

5.1.3 Edit ACL Structure

The ACL definition can be split into two big parts, *Match settings* and *Change settings*. In the matching sections the ACLs contain attributes that has to be met in order to use the ACL. If the attributes defined in the ACL do not match with the attributes that are sent, then the ACL does not take any affect, but any other match ACL will. The change sections contain the rules to reduce the possible options for a ticket.

Match settings

`Properties` This section contains matching options that can be changed on the fly. For example on a ticket creation time the data of the ticket changes dynamically as the agent sets the information. If an ACL is set to match a ticket attribute then only when the matching attribute is selected the ACL will be active and might reduce other ticket attributes, but as soon as another value is selected the ACL will not take any affect.

`PropertiesDatabase` This section is similar to `Properties` but does not take changes in ticket attributes that are not saved into the database, this means that changing an attribute without submit will not make any effect. This section is not use for ticket creation screens (as tickets are not yet created in the database).

Change settings

`Possible` This section is used to reset the data to be reduce to only the elements that are set in this section.

`PossibleAdd` This section is used to add missing elements that were reduced in other ACLs. This section is only used in together with other ACLs that have `Possible` or `PossibleNot` sections.

`PossibleNot` This section is used to remove specific elements from the current data. It could be used stand alone or together with other ACLs with a `Possible` or `PossibleAdd` sections.

Modifiers

In order to make the development of ACLs easier and more powerful there is a set of so called modifiers for the attributes on each section. This modifiers are explained below:

[Not] This modifier is used to negate a value, for example [Not]2 low. Talking about priorities will be the same as to have: *1 very low*, *3 normal*, *4 high*, *5 very high*.

[RegExp] It is used to define a regular expression for matching several values, for example [RegExp]low. In this case talking about priorities is the same as *1 very low*, *2 low*.

[regexp] It is very similar to [RegExp] but it is case insensitive.

[NotRegExp] Negated regular expressions, for example [NotRegExp]low. Talking about priorities is the same as *3 normal*, *4 high*, *5 very high*.

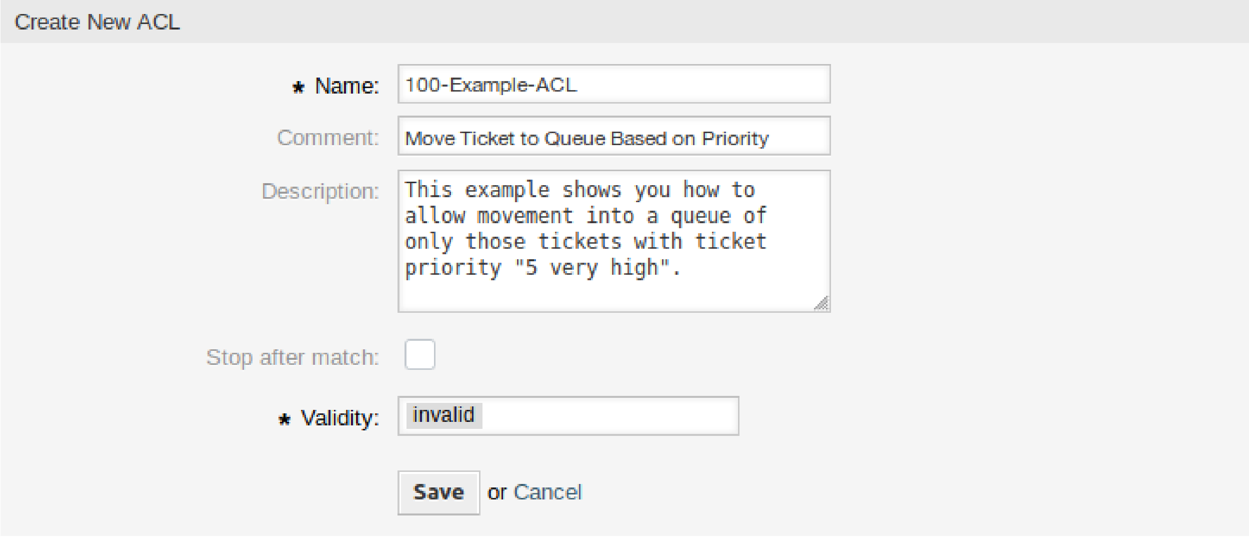
[Notregexp] It is very similar to [NotRegExp] but it is case insensitive.

5.1.4 ACL Examples

Move Ticket to Queue Based on Priority

This example shows you how to allow movement into a queue of only those tickets with ticket priority *5 very high*.

First, it needs to have a name. In this case, it is *100-Example-ACL*. Note that the ACLs will be numerically sorted before execution, so you should use the names carefully. The comment and the description fields are optional.



Create New ACL

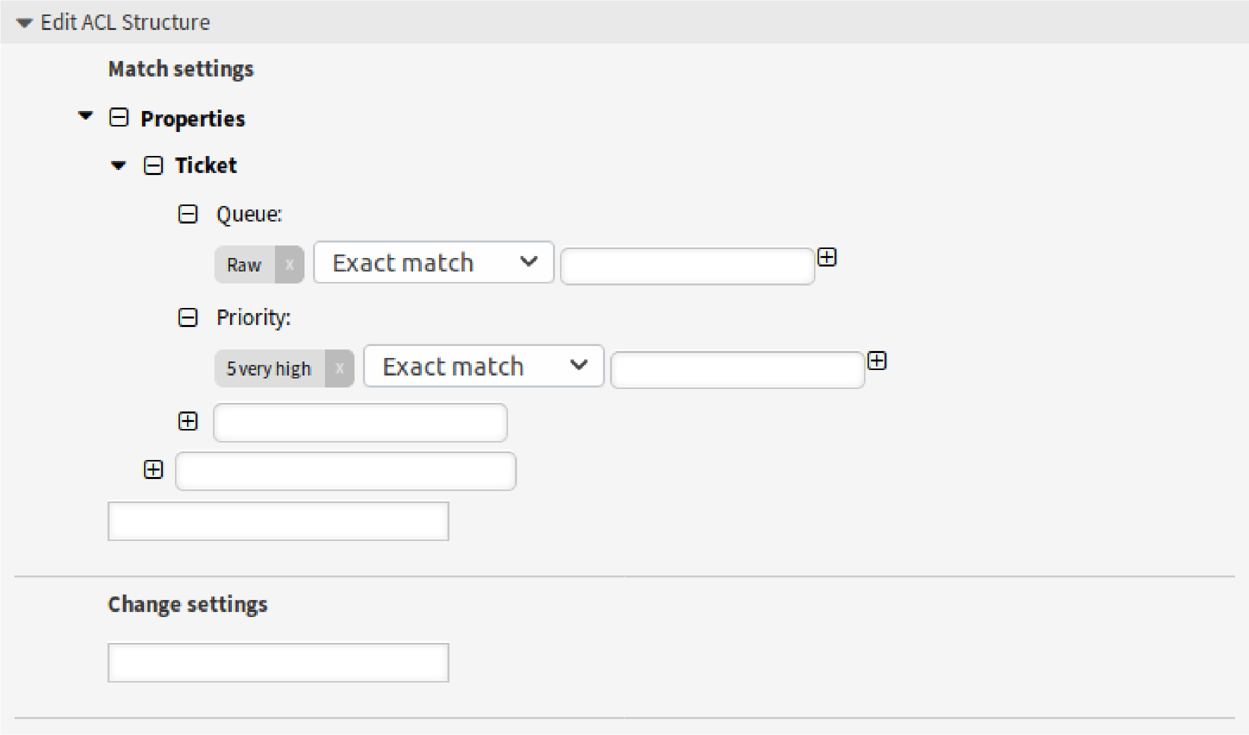
* Name:	100-Example-ACL
Comment:	Move Ticket to Queue Based on Priority
Description:	This example shows you how to allow movement into a queue of only those tickets with ticket priority "5 very high".
Stop after match:	□
* Validity:	invalid

Fig. 4: 100-Example-ACL - Basic Settings

Secondly, you have a *Properties* section which is a filter for your tickets. All the criteria defined here will be applied to a ticket to determine if the ACL must be applied or not. In our example, a ticket will match if it is in the queue *Raw* and has priority *5 very high*. This is also affected by changes in the form (e. g. if the ticket is the queue *Raw* and had a priority *3 normal* at this moment the ACL will not match, but then priority drop-down is selected and the priority is changed now to *5 very high* then will also match).

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

Note: Don' t forget to set *Validity* to *valid* and deploy the newly created ACL.



Edit ACL Structure

Match settings

Properties

Ticket

Queue:

Raw x | Exact match

Priority:

5 very high x | Exact match

Change settings

Fig. 5: 100-Example-ACL - Match Settings

Move Ticket to Queue Based on Priority Stored in Database

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

Disable Ticket Close in Queue and Hide Close Ticket Action

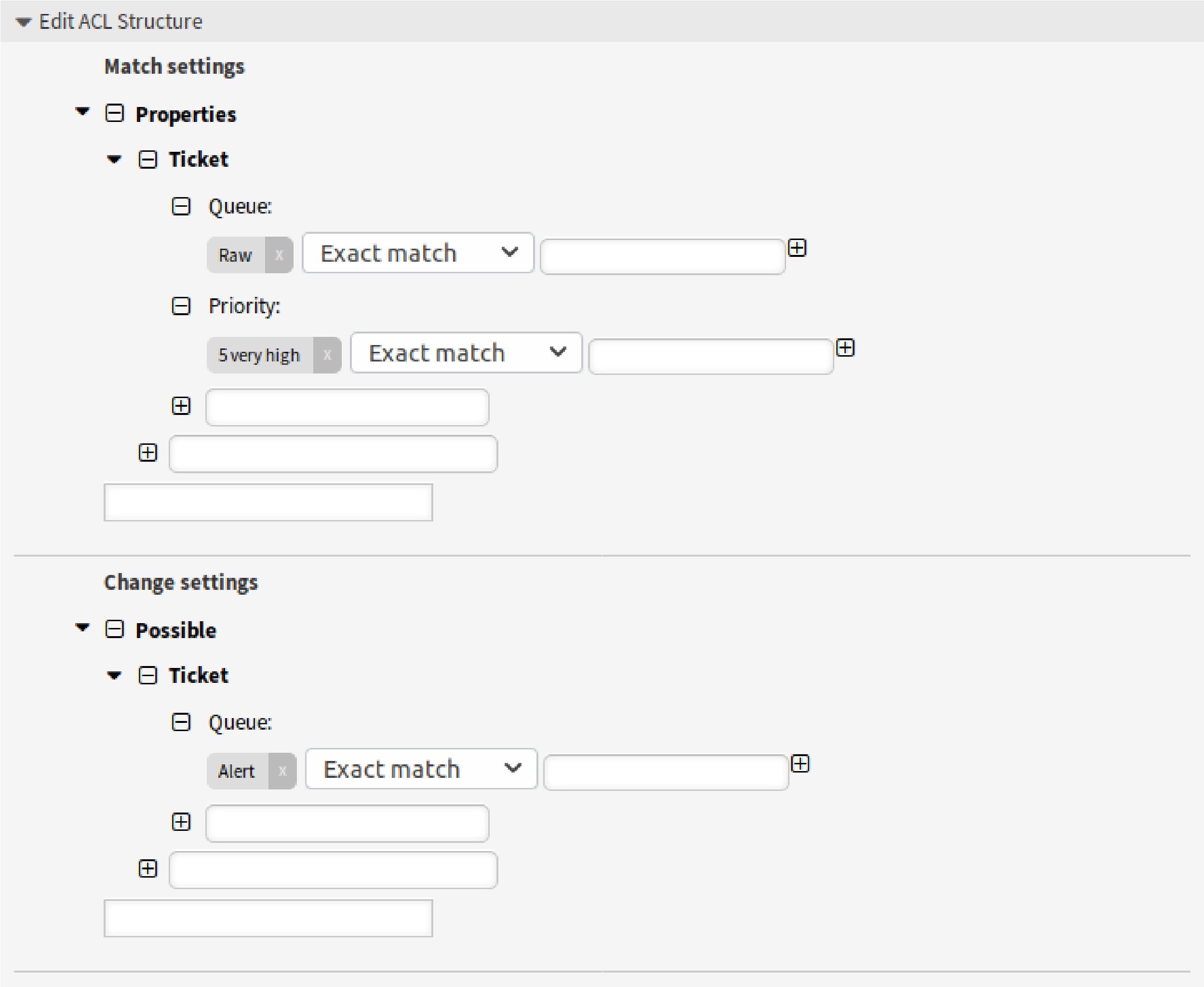
This example shows how to disable the closing of tickets in the queue *Raw*, and hide the *Close Ticket* action. It is possible to filter a ticket field (state) with more than one possible values to select from. It is also possible to limit the actions that can be executed for a certain ticket. In this case, the ticket cannot be closed.

Remove State

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

Using Regular Expressions

This example shows you how you can use regular expressions for matching tickets and for filtering the available options. This ACL only shows *Hardware* services for tickets that are created in queues that start with *HW*.



Edit ACL Structure

Match settings

Properties

Ticket

Queue:

Raw Exact match

Priority:

5 very high Exact match

Change settings

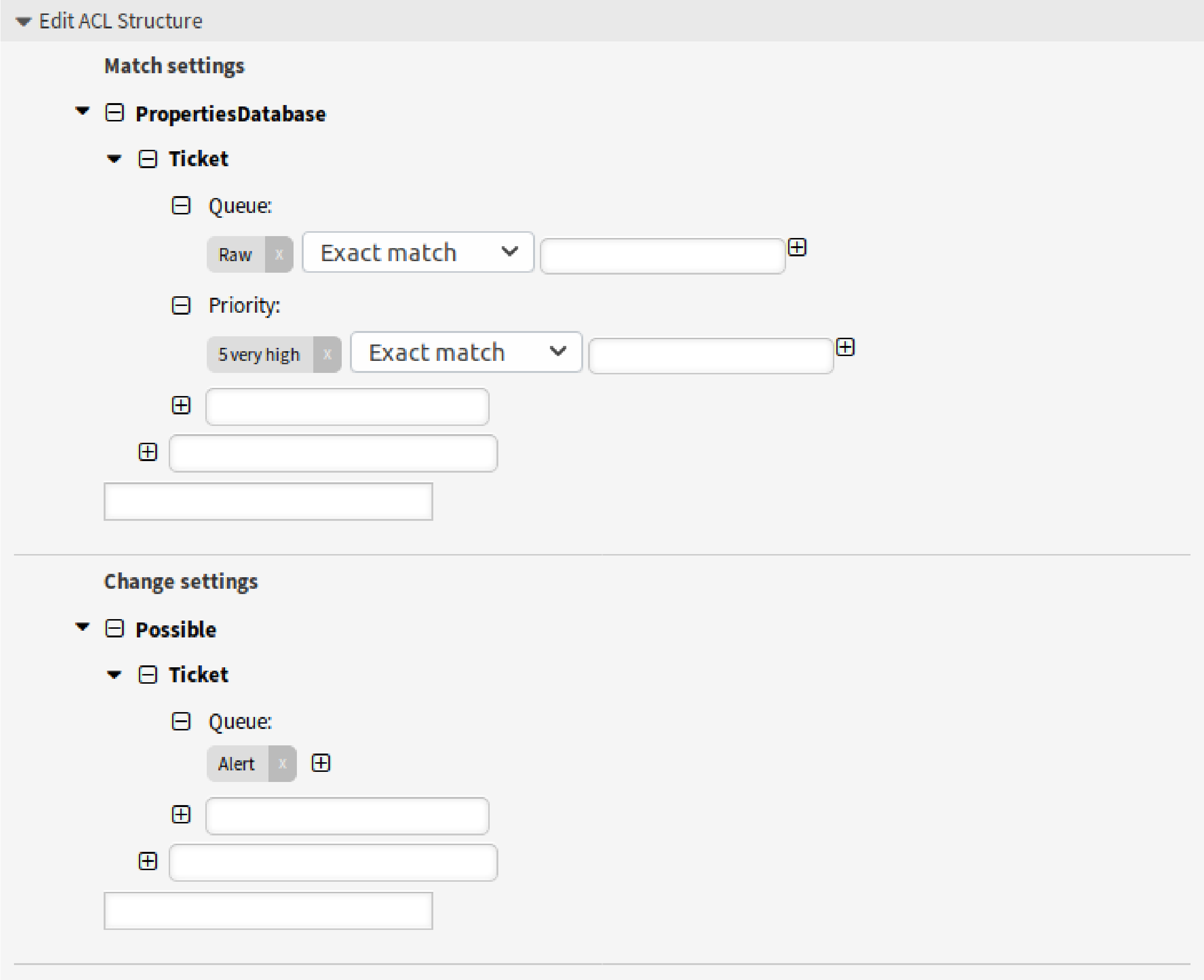
Possible

Ticket

Queue:

Alert Exact match

Fig. 6: 100-Example-ACL - Change Settings



Edit ACL Structure

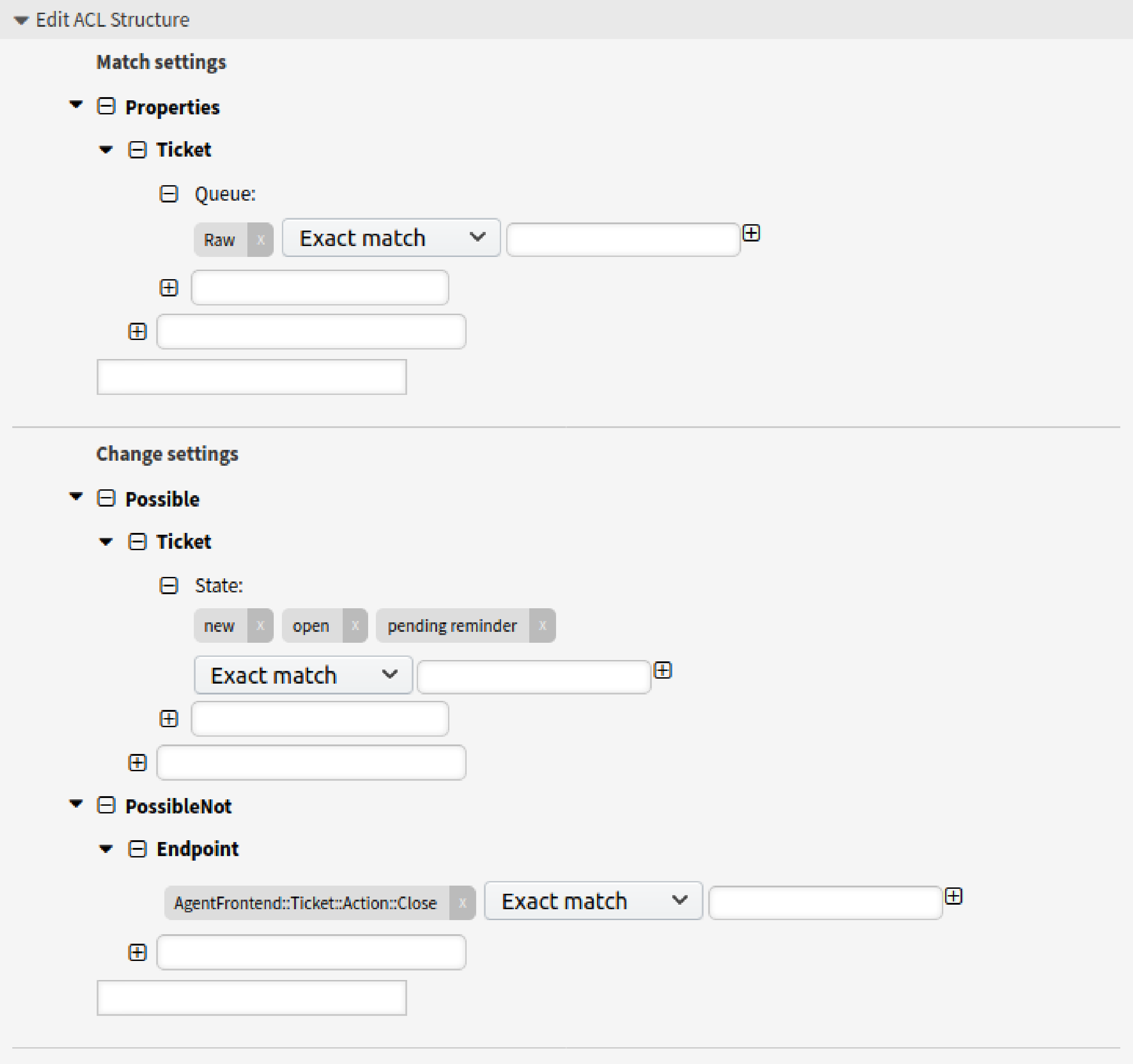
Match settings

- PropertiesDatabase
  - Ticket
    - Queue: Raw — Exact match
    - Priority: 5 very high — Exact match

Change settings

- Possible
  - Ticket
    - Queue: Alert

Fig. 7: 101-Example-ACL



Edit ACL Structure

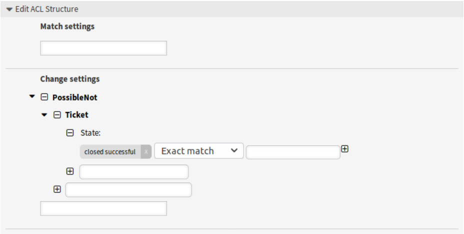
Match settings

- Properties
  - Ticket
    - Queue:
      - Raw — Exact match

Change settings

- Possible
  - Ticket
    - State:
      - new, open, pending reminder — Exact match
- PossibleNot
  - Endpoint
    - AgentFrontend::Ticket::Action::Close — Exact match

Fig. 8: 102-Example-ACL



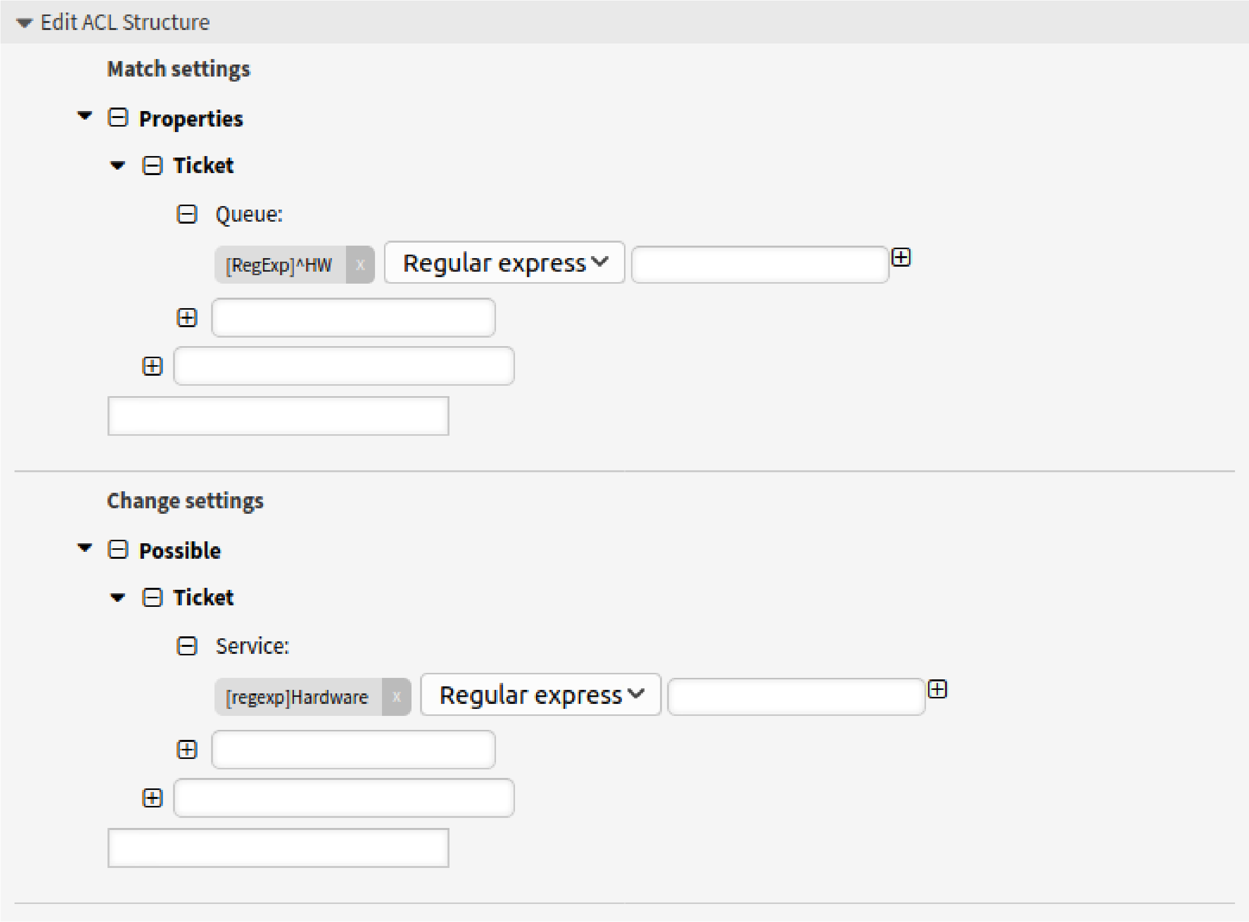
Edit ACL Structure

Match settings

Change settings

- PossibleNot
  - Ticket
    - State:
      - closed successful — Exact match

Fig. 9: 103-Example-ACL



Edit ACL Structure

Match settings

Properties

Ticket

Queue:

[RegExp]^HW

Regular express

Change settings

Possible

Ticket

Service:

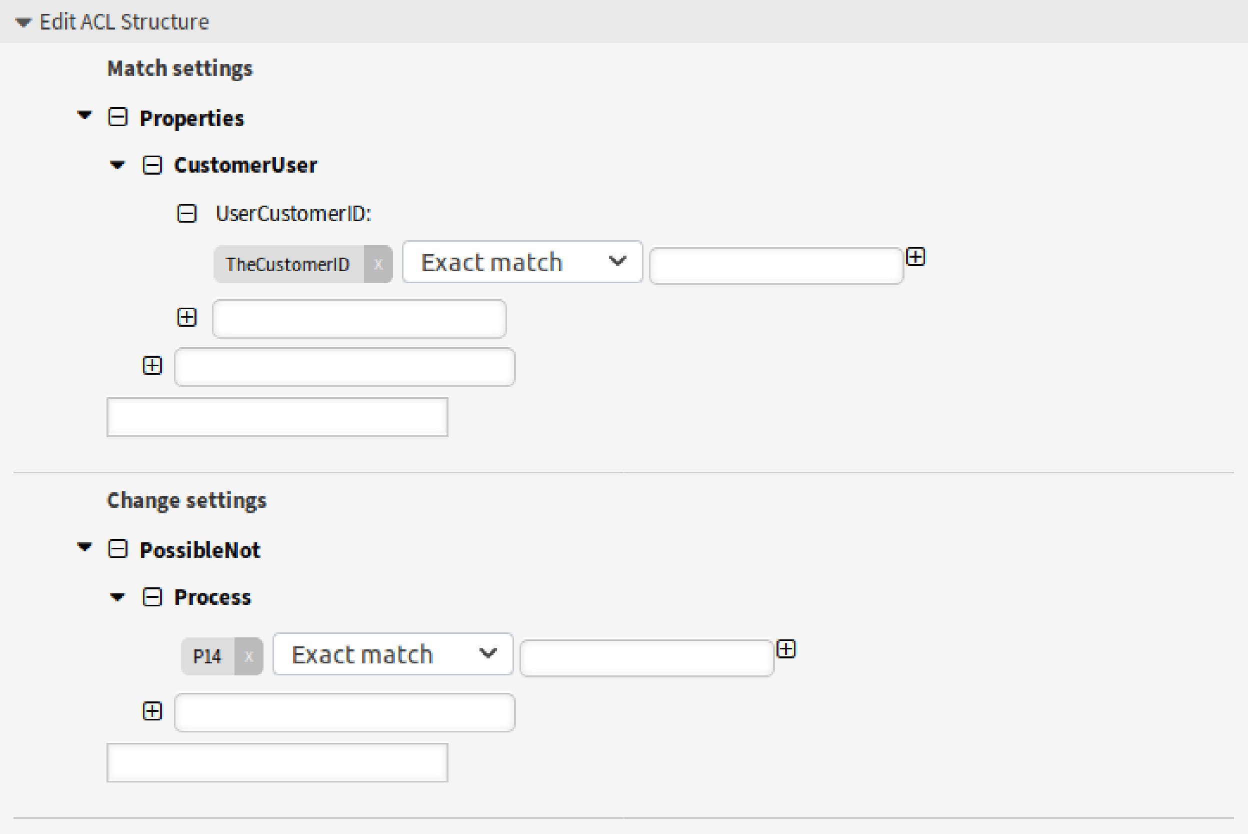
[regexp]Hardware

Regular express

Fig. 10: 104-Example-ACL

Disallow Process For Customer

This ACL restricts the process *P14* in the external interface using the customer ID *TheCustomerID*.



Edit ACL Structure

Match settings

Properties

CustomerUser

UserCustomerID:

TheCustomerID — Exact match

Change settings

PossibleNot

Process

P14 — Exact match

Fig. 11: 105-Example-ACL

5.1.5 ACL Reference

Properties, keys and values that can be used in ACLs are highly depend on the OTRS installation. For example the possibilities can be extended by installing extension modules, as well as it can be depend on the customer user mapping set in `Config.pm`. Therefore it is not possible to provide a full ACL reference, that contains all settings.

For properties, keys and values that can be used in ACLs, see the following example ACL in YAML format.

```yaml
---
- ChangeBy: root@localhost
  ChangeTime: 2020-04-15 16:46:23
  Comment: ACL Reference.
  ConfigMatch:
    Properties:
      # Match properties (current values from the form).
      CustomerUser:
        UserLogin:
        - some login
```

(continues on next page)

(continued from previous page)

```
        UserCustomerID:
        - some customer ID
        Group_rw:
        - some group
      DynamicField:
        # Names must be in DynamicField_<field_name> format.
        # Values for dynamic fields must always be the untranslated internal
        #   data keys specified in the dynamic field definition and not the
        #   data values shown to the user.
        DynamicField_Field1:
        - some value
        DynamicField_OtherField:
        - some value
        DynamicField_TicketFreeText2:
        - some value
        # more dynamic fields
      Frontend:
        Endpoint:
        - AgentFrontend::KnowledgeBaseArticleDetailView::Property
        - AgentFrontend::PersonalPreferences
        - AgentFrontend::ProcessTicketNextStep
        - AgentFrontend::Ticket::Action::Close
        - AgentFrontend::Ticket::Action::Customer
        - AgentFrontend::Ticket::Action::EmailOutbound
        - AgentFrontend::Ticket::Action::FreeText
        - AgentFrontend::Ticket::Action::Link
        - AgentFrontend::Ticket::Action::Lock
        - AgentFrontend::Ticket::Action::Merge
        - AgentFrontend::Ticket::Action::Move
        - AgentFrontend::Ticket::Action::Note
        - AgentFrontend::Ticket::Action::Owner
        - AgentFrontend::Ticket::Action::Pending
        - AgentFrontend::Ticket::Action::PhoneCallInbound
        - AgentFrontend::Ticket::Action::PhoneCallOutbound
        - AgentFrontend::Ticket::Action::Print
        - AgentFrontend::Ticket::Action::Priority
        - AgentFrontend::Ticket::Action::Redirect
        - AgentFrontend::Ticket::Action::Responsible
        - AgentFrontend::Ticket::Action::SmsOutbound
        - AgentFrontend::Ticket::Action::TicketHistory
        - AgentFrontend::Ticket::Action::Unlock
        - AgentFrontend::Ticket::Action::Unwatch
        - AgentFrontend::Ticket::Action::Watch
        - AgentFrontend::TicketArticle::Action::CopyLink
        - AgentFrontend::TicketArticle::Action::Forward
        - AgentFrontend::TicketArticle::Action::MarkAsImportant
        - AgentFrontend::TicketArticle::Action::Plain
        - AgentFrontend::TicketArticle::Action::Print
        - AgentFrontend::TicketArticle::Action::Redirect
        - AgentFrontend::TicketArticle::Action::Reply
        - AgentFrontend::TicketArticle::Action::ReplyAll
        - AgentFrontend::TicketArticle::Action::ReplyToNote
```

(continues on next page)

(continued from previous page)

```
        - AgentFrontend::TicketArticle::Action::ReplyViaSms
        - AgentFrontend::TicketArticle::Action::Split
        - AgentFrontend::TicketArticle::Action::UnmarkAsImportant
        - AgentFrontend::TicketCreateEmail
        - AgentFrontend::TicketCreatePhone
        - AgentFrontend::TicketCreateProcess
        - AgentFrontend::TicketCreateSMS
        - AgentFrontend::TicketDetailView::Property
        - AgentFrontend::TicketList::Bulk
        - AgentFrontend::TicketList::Filter
        - ...
        - ExternalFrontend::PersonalPreferences
        - ExternalFrontend::ProcessTicketCreate
        - ExternalFrontend::ProcessTicketNextStep
        - ExternalFrontend::TicketCreate
        - ExternalFrontend::TicketDetailView
        - ...
      Owner:
        UserLogin:
        - some login
        Group_rw:
        - some group
        Role:
        - admin
        # more owner attributes
      Priority:
        ID:
        - some ID
        Name:
        - some name
        # more priority attributes
      Process:
        ProcessEntityID:
        # the process that the current ticket is part of
        - Process-9c378d7cc59f0fce4cee7bb9995ee3eb
        ActivityEntityID:
        # the current activity of the ticket
        - Activity-f8b2fdebe54eeb7b147a5f8e1da5e35c
        ActivityDialogEntityID:
        # the current activity dialog that the agent/customer is using
        - ActivityDialog-aff0ae05fe6803f38de8fff6cf33b7ce
      Queue:
        Name:
        - Raw
        QueueID:
        - some ID
        GroupID:
        - some ID
        Email:
        - some email
        RealName:
        - OTRS System
```

(continues on next page)

(continued from previous page)

```
        # more queue attributes
      Responsible:
        UserLogin:
        - some login
        Group_rw:
        - some group
        Role:
        - admin
        # more responsible attributes
      Service:
        ServiceID:
        - some ID
        Name:
        - some name
        ParentID:
        - some ID
        # more service attributes
      SLA:
        SLAID:
        - some ID
        Name:
        - some name
        Calendar:
        - some calendar
        # more SLA attributes
      State:
        ID:
        - some ID
        Name:
        - some name
        TypeName:
        - some state type name
        TypeID:
        - some state type ID
        # more state attributes
      Ticket:
        Queue:
        - Raw
        State:
        - new
        - open
        Priority:
        - some priority
        Lock:
        - lock
        CustomerID:
        - some ID
        CustomerUserID:
        - some ID
        Owner:
        - some owner
        DynamicField_Field1:
```

(continues on next page)

(continued from previous page)

```
        - some value
        DynamicField_MyField:
        - some value
        # more ticket attributes
      Type:
        ID:
        - some ID
        Name:
        - some name
        # more type attributes
      User:
        UserLogin:
        - some_login
        Group_rw:
        - some group
        Role:
        - admin
    PropertiesDatabase:
      # Match properties (existing values from the database).
      # Please note that Frontend is not in the database, but in the framework.
      # See section "Properties", the same configuration can be used here.
  ConfigChange:
    Possible:
      # Reset possible options (white list).
      Action:
      # Possible action options (white list).
      - ...
      ActivityDialog:
      # Limit the number of possible activity dialogs the agent/customer can use in a process ticket.
      - ActivityDialog-aff0ae05fe6803f38de8fff6cf33b7ce
      - ActivityDialog-429d61180a593414789a8087cc4b3c6f
      - ...
      Endpoint:
      # Limit the functions on agent interface.
      - AgentFrontend::KnowledgeBaseArticleDetailView::Property
      - AgentFrontend::PersonalPreferences
      - AgentFrontend::ProcessTicketNextStep
      - AgentFrontend::Ticket::Action::Close
      - AgentFrontend::Ticket::Action::Customer
      - AgentFrontend::Ticket::Action::EmailOutbound
      - AgentFrontend::Ticket::Action::FreeText
      - AgentFrontend::Ticket::Action::Merge
      - AgentFrontend::Ticket::Action::Move
      - AgentFrontend::Ticket::Action::Note
      - AgentFrontend::Ticket::Action::Owner
      - AgentFrontend::Ticket::Action::Pending
      - AgentFrontend::Ticket::Action::PhoneCallInbound
      - AgentFrontend::Ticket::Action::PhoneCallOutbound
      - AgentFrontend::Ticket::Action::Priority
      - AgentFrontend::Ticket::Action::Redirect
```

(continues on next page)

(continued from previous page)

```
      - AgentFrontend::Ticket::Action::Responsible
      - AgentFrontend::Ticket::Action::SmsOutbound
      - AgentFrontend::TicketArticle::Action::CopyLink
      - AgentFrontend::TicketArticle::Action::Forward
      - AgentFrontend::TicketArticle::Action::Redirect
      - AgentFrontend::TicketArticle::Action::Reply
      - AgentFrontend::TicketArticle::Action::ReplyAll
      - AgentFrontend::TicketArticle::Action::ReplyToNote
      - AgentFrontend::TicketArticle::Action::ReplyViaSms
      - AgentFrontend::TicketArticle::Action::Split
      - AgentFrontend::TicketCreateEmail
      - AgentFrontend::TicketCreatePhone
      - AgentFrontend::TicketCreateProcess
      - AgentFrontend::TicketCreateSMS
      - AgentFrontend::TicketDetailView::Property
      - AgentFrontend::TicketList::Bulk
      - AgentFrontend::TicketList::Filter
      - ...
      # Limit the functions on external interface.
      - ExternalFrontend::PersonalPreferences
      - ExternalFrontend::ProcessTicketCreate
      - ExternalFrontend::ProcessTicketNextStep
      - ExternalFrontend::TicketCreate
      - ExternalFrontend::TicketDetailView
      - ...
      Process:
      # Limit the number of possible processes that can be started.
      - Process-9c378d7cc59f0fce4cee7bb9995ee3eb
      - Process-12345678901234567890123456789012
      - ...
      Ticket:
      # Possible ticket options (white list).
        Queue:
        - Raw
        - some other queue
        State:
        - some state
        Priority:
        - 5 very high
        DynamicField_Field1:
        - some value
        DynamicField_MyField:
        - some value
        # more dynamic fields
        NewOwner:
        # For ticket action screens, where the Owner is already set.
        - some owner
        OldOwner:
        # For ticket action screens, where the Owner is already set.
        - some owner
        Owner:
        # For ticket create screens, because Owner is not set yet.
```

(continues on next page)

(continued from previous page)

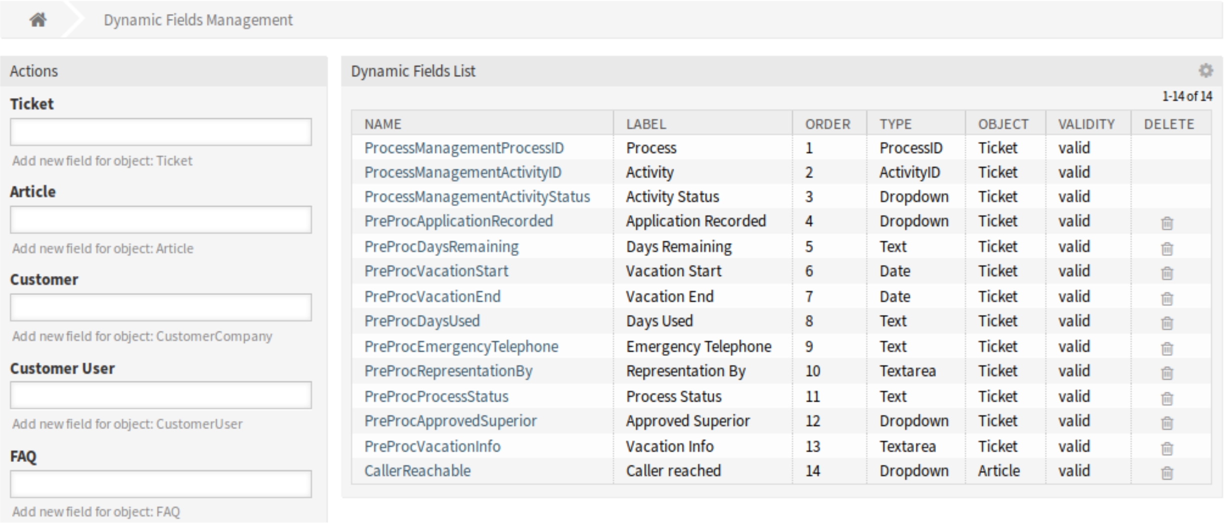
```
      - some owner
      # more ticket attributes
  PossibleAdd:
     # Add options (white list).
     # See section "Possible", the same configuration can be used here.
  PossibleNot:
     # Remove options (black list).
     # See section "Possible", the same configuration can be used here.
CreateBy: root@localhost
CreateTime: 2020-04-15 16:46:23
Description: This is the long description of the ACL to explain its usage.
ID: 1
Name: 200-ACL-Reference
StopAfterMatch: 0
ValidID: 3
```

5.2 Dynamic Fields

Beside general information that required for all tickets, organizations have individual needs to add specific details to tickets. This needed information takes various formats like texts, integers, date-time and more.

OTRS supports adding a so-called *dynamic field* to handle texts, integers, drop-down lists, multi-select fields, date-time, checkboxes and more. OTRS administrators can define where those fields should be visible or editable, and of course, the dynamic fields are also available in statistics and reports.

Use this screen to manage dynamic fields in the system. A fresh OTRS installation contains three dynamic fields by default. The dynamic field management screen is available in the *Dynamic Fields* module of the *Processes & Automation* group.



Dynamic Fields Management

Actions

Ticket

Add new field for object: Ticket

Article

Add new field for object: Article

Customer

Add new field for object: CustomerCompany

Customer User

Add new field for object: CustomerUser

FAQ

Add new field for object: FAQ

Dynamic Fields List

1-14 of 14

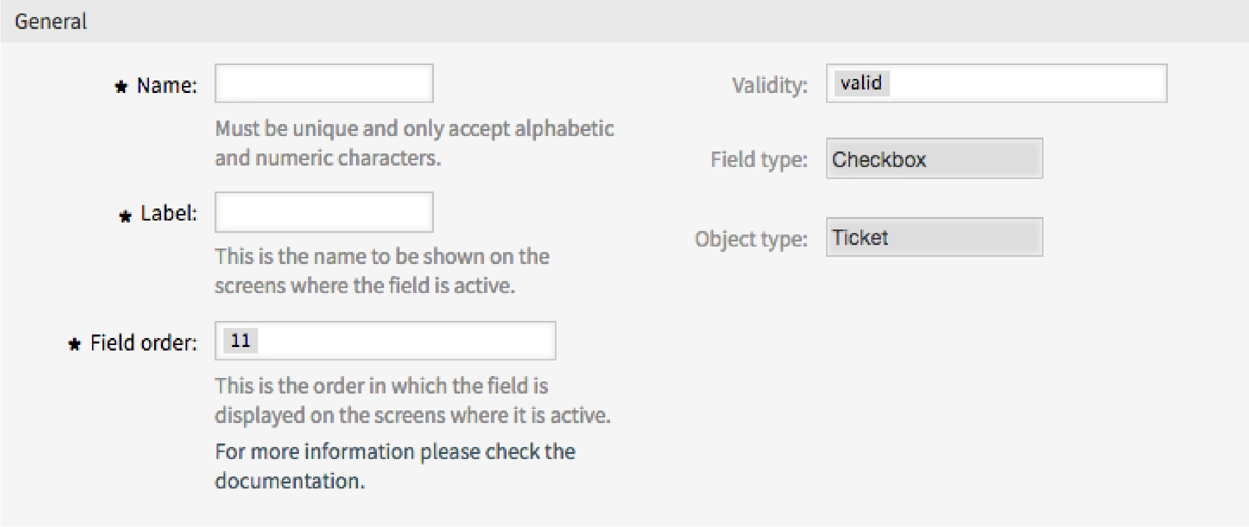
NAME	LABEL	ORDER	TYPE	OBJECT	VALIDITY	DELETE
ProcessManagementProcessID	Process	1	ProcessID	Ticket	valid	
ProcessManagementActivityID	Activity	2	ActivityID	Ticket	valid	
ProcessManagementActivityStatus	Activity Status	3	Dropdown	Ticket	valid	
PreProcApplicationRecorded	Application Recorded	4	Dropdown	Ticket	valid	🗑
PreProcDaysRemaining	Days Remaining	5	Text	Ticket	valid	🗑
PreProcVacationStart	Vacation Start	6	Date	Ticket	valid	🗑
PreProcVacationEnd	Vacation End	7	Date	Ticket	valid	🗑
PreProcDaysUsed	Days Used	8	Text	Ticket	valid	🗑
PreProcEmergencyTelephone	Emergency Telephone	9	Text	Ticket	valid	🗑
PreProcRepresentationBy	Representation By	10	Textarea	Ticket	valid	🗑
PreProcProcessStatus	Process Status	11	Text	Ticket	valid	🗑
PreProcApprovedSuperior	Approved Superior	12	Dropdown	Ticket	valid	🗑
PreProcVacationInfo	Vacation Info	13	Textarea	Ticket	valid	🗑
CallerReachable	Caller reached	14	Dropdown	Article	valid	🗑

Fig. 12: Dynamic Field Management Screen

5.2.1 Manage Dynamic Fields

To create a new dynamic field:

1. Choose an object in the left sidebar and select a dynamic field type from its drop-down.
2. Fill in the required fields.
3. Click on the *Save* button.



General

★ Name:

Must be unique and only accept alphabetic and numeric characters.

★ Label:

This is the name to be shown on the screens where the field is active.

★ Field order: 11

This is the order in which the field is displayed on the screens where it is active.

For more information please check the documentation.

Validity: valid

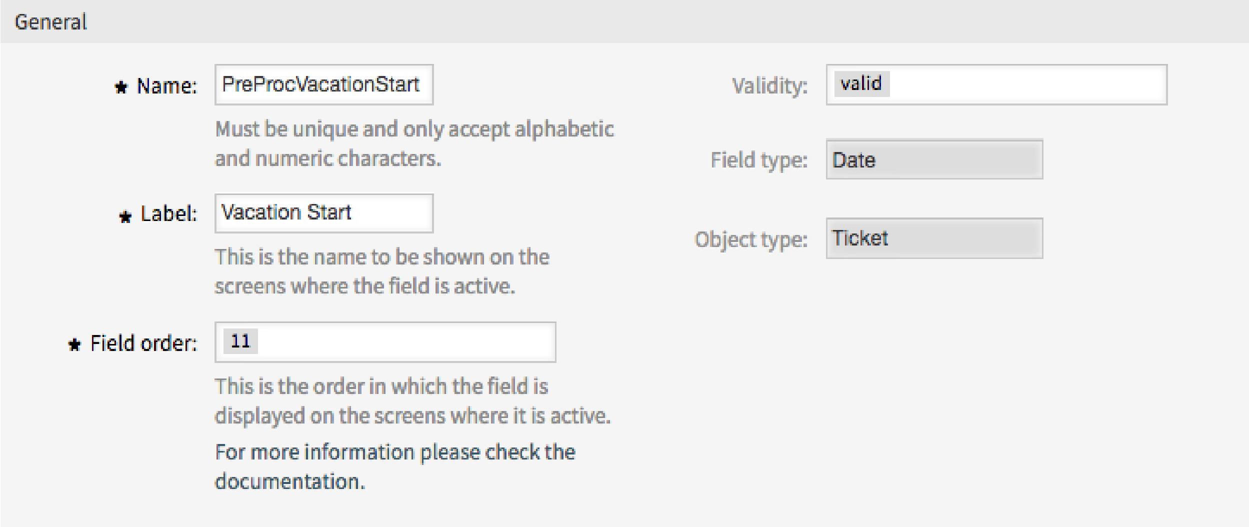
Field type: Checkbox

Object type: Ticket

Fig. 13: Create New Dynamic Field Screen

To edit a dynamic field:

1. Click on a dynamic field in the list of dynamic fields.
2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.



General

★ Name: PreProcVacationStart

Must be unique and only accept alphabetic and numeric characters.

★ Label: Vacation Start

This is the name to be shown on the screens where the field is active.

★ Field order: 11

This is the order in which the field is displayed on the screens where it is active.

For more information please check the documentation.

Validity: valid

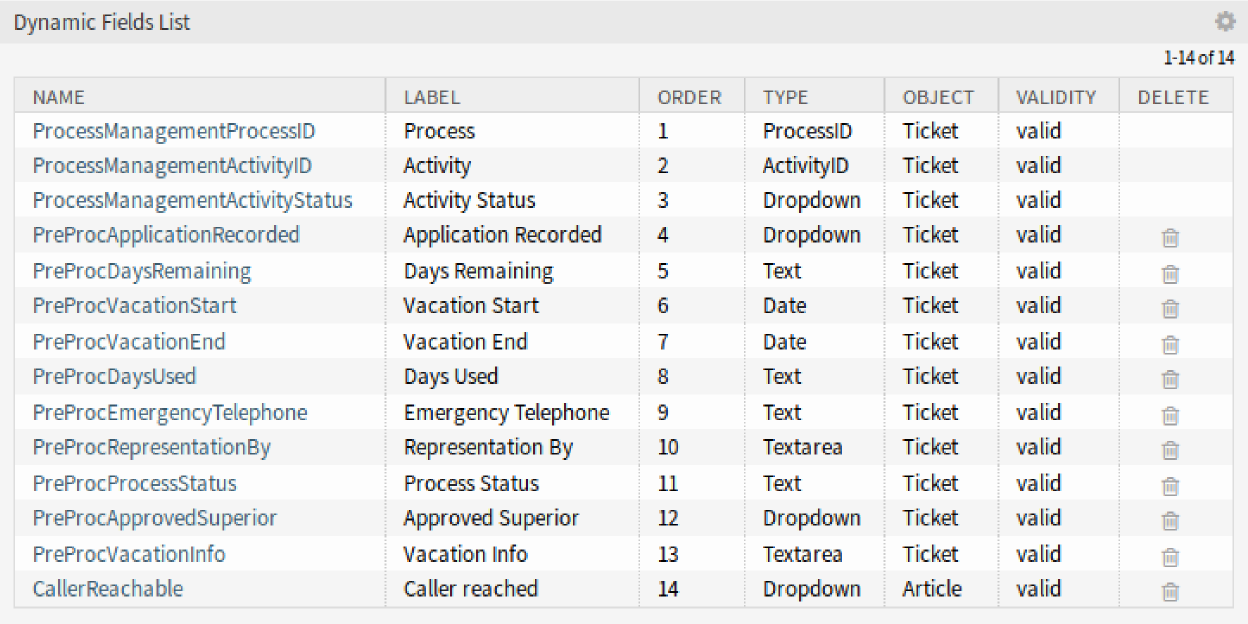
Field type: Date

Object type: Ticket

Fig. 14: Edit Dynamic Field Screen

To delete a dynamic field:

1. Click on the trash icon in the last column of the overview table.
2. Click on the *Confirm* button.



Dynamic Fields List

1-14 of 14

NAME	LABEL	ORDER	TYPE	OBJECT	VALIDITY	DELETE
ProcessManagementProcessID	Process	1	ProcessID	Ticket	valid	
ProcessManagementActivityID	Activity	2	ActivityID	Ticket	valid	
ProcessManagementActivityStatus	Activity Status	3	Dropdown	Ticket	valid	
PreProcApplicationRecorded	Application Recorded	4	Dropdown	Ticket	valid	🗑
PreProcDaysRemaining	Days Remaining	5	Text	Ticket	valid	🗑
PreProcVacationStart	Vacation Start	6	Date	Ticket	valid	🗑
PreProcVacationEnd	Vacation End	7	Date	Ticket	valid	🗑
PreProcDaysUsed	Days Used	8	Text	Ticket	valid	🗑
PreProcEmergencyTelephone	Emergency Telephone	9	Text	Ticket	valid	🗑
PreProcRepresentationBy	Representation By	10	Textarea	Ticket	valid	🗑
PreProcProcessStatus	Process Status	11	Text	Ticket	valid	🗑
PreProcApprovedSuperior	Approved Superior	12	Dropdown	Ticket	valid	🗑
PreProcVacationInfo	Vacation Info	13	Textarea	Ticket	valid	🗑
CallerReachable	Caller reached	14	Dropdown	Article	valid	🗑

Fig. 15: Delete Dynamic Field Screen

Note: If several dynamic fields are added to the system, use the filter box to find a particular dynamic field by just typing the name to filter.

5.2.2 Dynamic Field Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

General Dynamic Field Settings

These settings are the same for all types of dynamic fields.

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

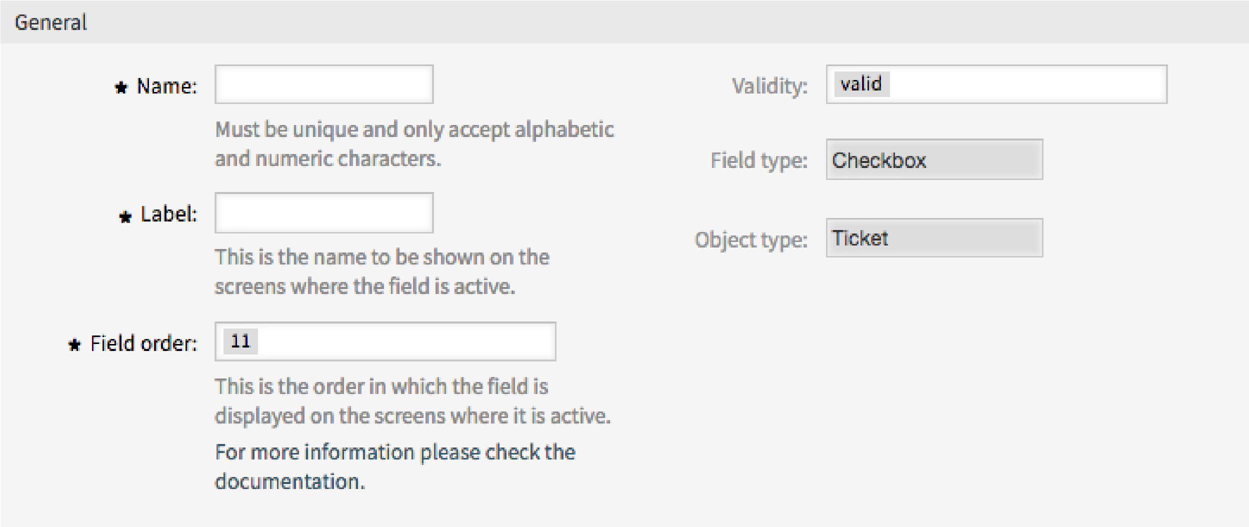
Label * This is the name to be shown on the screens where the field is active.

See also:

It is possible to add translations for a dynamic field label. Label translations have to be added manually to language translation files.

Field order * This is the order in which the field is displayed on the screens where it is active.

Note: The configured value for the field order is considered in the following screens:



General

★ Name:

Must be unique and only accept alphabetic and numeric characters.

★ Label:

This is the name to be shown on the screens where the field is active.

★ Field order: 11

This is the order in which the field is displayed on the screens where it is active.

For more information please check the documentation.

Validity: valid

Field type: Checkbox

Object type: Ticket

Fig. 16: Dynamic Field General Screen

- all screens in the administrator interface
- all screens in the external interface

In other screens the order can be configured in other ways, e.g. via *a specific order of fields in the form configuration*.

Validity * Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.

Field type This type have been selected in the previous page and can not be changed here anymore. This is a read-only field.

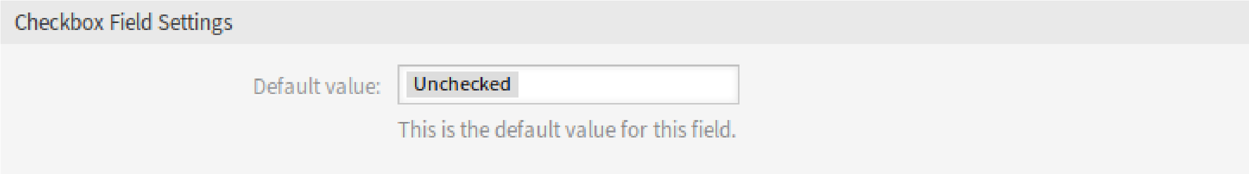
Object type This type have been selected in the previous page and can not be changed here anymore. This is a read-only field.

Note: The object type determines where the dynamic field can be used. For example dynamic field with object type *Ticket* can be used only in tickets, and can not be used in articles.

The following settings are relevant only for the particular type of dynamic fields.

Checkbox Dynamic Field Settings

Checkbox dynamic field is used to store true or false value.



Checkbox Field Settings

Default value: Unchecked

This is the default value for this field.

Fig. 17: Checkbox Dynamic Field Settings

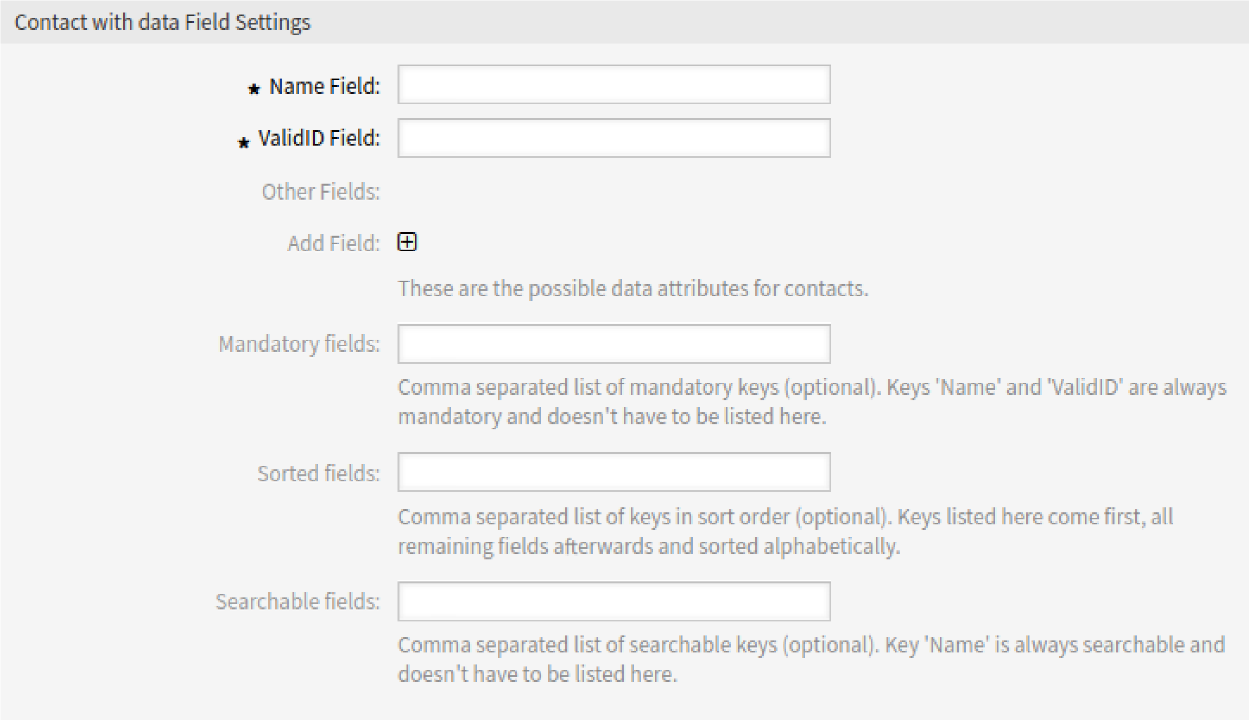
Default value * The default value for the checkbox.

Checked The checkbox is checked by default.

Unchecked The checkbox is unchecked by default.

Contact With Data Dynamic Field Settings

This dynamic field allows to add contacts with data to tickets.



Contact with data Field Settings

★ Name Field:

★ ValidID Field:

Other Fields:

Add Field: ⊞

These are the possible data attributes for contacts.

Mandatory fields:

Comma separated list of mandatory keys (optional). Keys 'Name' and 'ValidID' are always mandatory and doesn't have to be listed here.

Sorted fields:

Comma separated list of keys in sort order (optional). Keys listed here come first, all remaining fields afterwards and sorted alphabetically.

Searchable fields:

Comma separated list of searchable keys (optional). Key 'Name' is always searchable and doesn't have to be listed here.

Fig. 18: Contact With Data Dynamic Field Settings

Name Field * The attribute `Name` is always mandatory and it is not automatically added, so for each new data source this attribute must be added manually. Within the data source definition (or dynamic field configuration) this must be represented by the key `Name` and the value could be *Name* for example.

ValidID Field * The attribute `ValidID` is always mandatory and it is not automatically added, so for each new data source this attribute must be added manually. Within the data source definition (or dynamic field configuration) this must be represented by the key `ValidID` and the value could be *Validity* for example.

Other Fields These are the possible data attributes for contacts. Clicking on the button of the *Add Fields* will add two new fields, where a *Key* (internal value) and a *Value* (displayed value) can be set. With the button you can add multiple key-value pairs.

> **Warning:** The keys `Name` and `ValidID` are already used by *Name Field* and *ValidID Field*. Do not use these keys again!

Add Fields Use this button to add more fields to dynamic field.

Mandatory fields Comma separated list of mandatory keys.

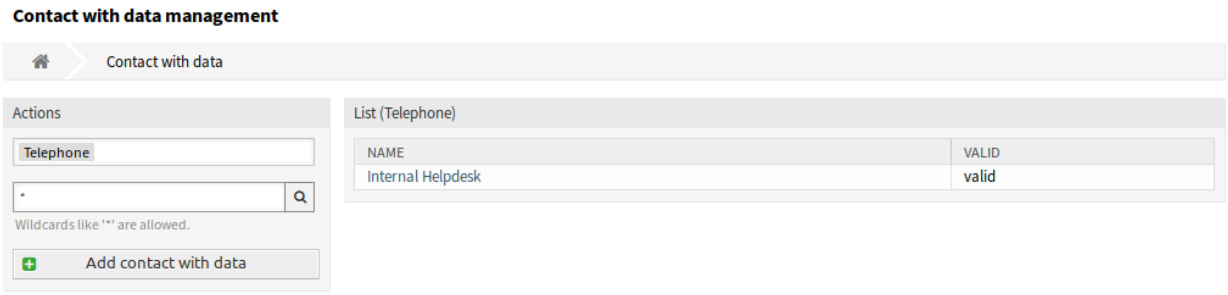
Note: Keys `Name` and `ValidID` are always mandatory and doesn't have to be listed here.

Sorted fields Comma separated list of keys in sort order. Keys listed here come first, all remaining fields afterwards and sorted alphabetically.

Searchable fields Comma separated list of searchable keys.

Note: Key `Name` is always searchable and doesn't have to be listed here.

When the dynamic field was saved, click on the name of the newly created dynamic field in the overview table. The *Edit Dynamic Field* screen will open. There is a button *Add or edit contacts*, that points to *Tickets* → *Edit contacts with data* to add some data.



NAME	VALID
Internal Helpdesk	valid

Fig. 19: Contact With Data Management Screen

To add a new contact with data:

1. Select a dynamic field from the drop-down list in *Actions* widget of the left sidebar.
2. Click on the *Add contact with data* button in the left sidebar.
3. Fill in the required fields.
4. Click on the *Save* button.

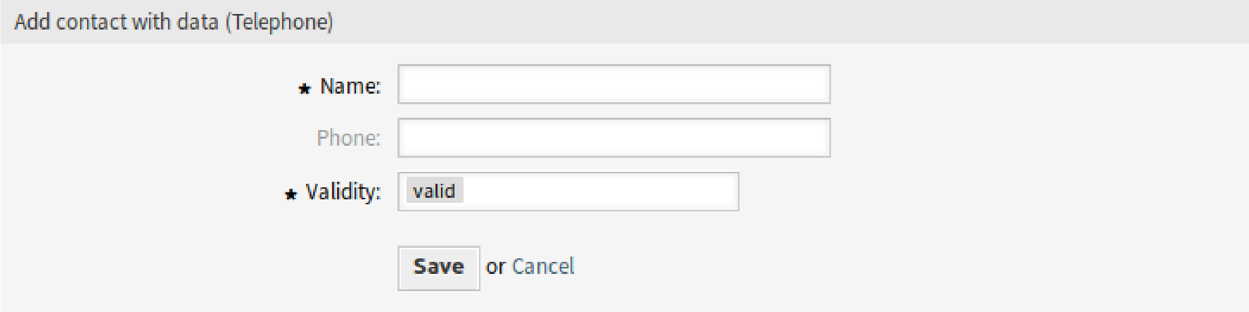
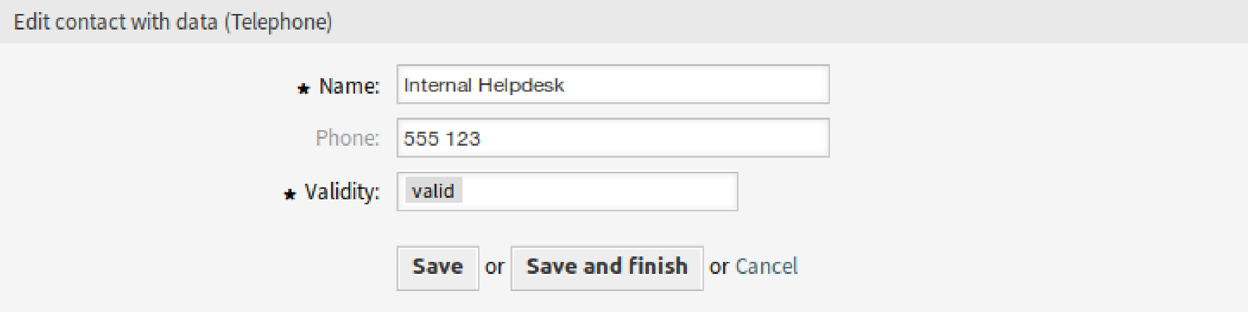


Fig. 20: Add Contact With Data Screen

To edit a contact with data:

1. Click on a contact with data in the list of contact with data entries.
2. Modify the fields.

3. Click on the *Save* or *Save and finish* button.



Edit contact with data (Telephone)

★ Name: Internal Helpdesk

Phone: 555 123

★ Validity: valid

Fig. 21: Edit Contact With Data Screen

The usage of this type of dynamic field is more complex then the others. 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).
   - Type *Name* into *Name Field*.
   - Type *Validity* into *ValidID Field*.
   - Add any other attribute with *Add Fields* button 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 *Manage Contacts with Data* screen from the main menu of the agent interface.
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 `Forms###AgentFrontend::TicketCreate::Phone::CreateProperties` and do the same for `AgentFrontend::TicketDetailView::WidgetType###Properties`.

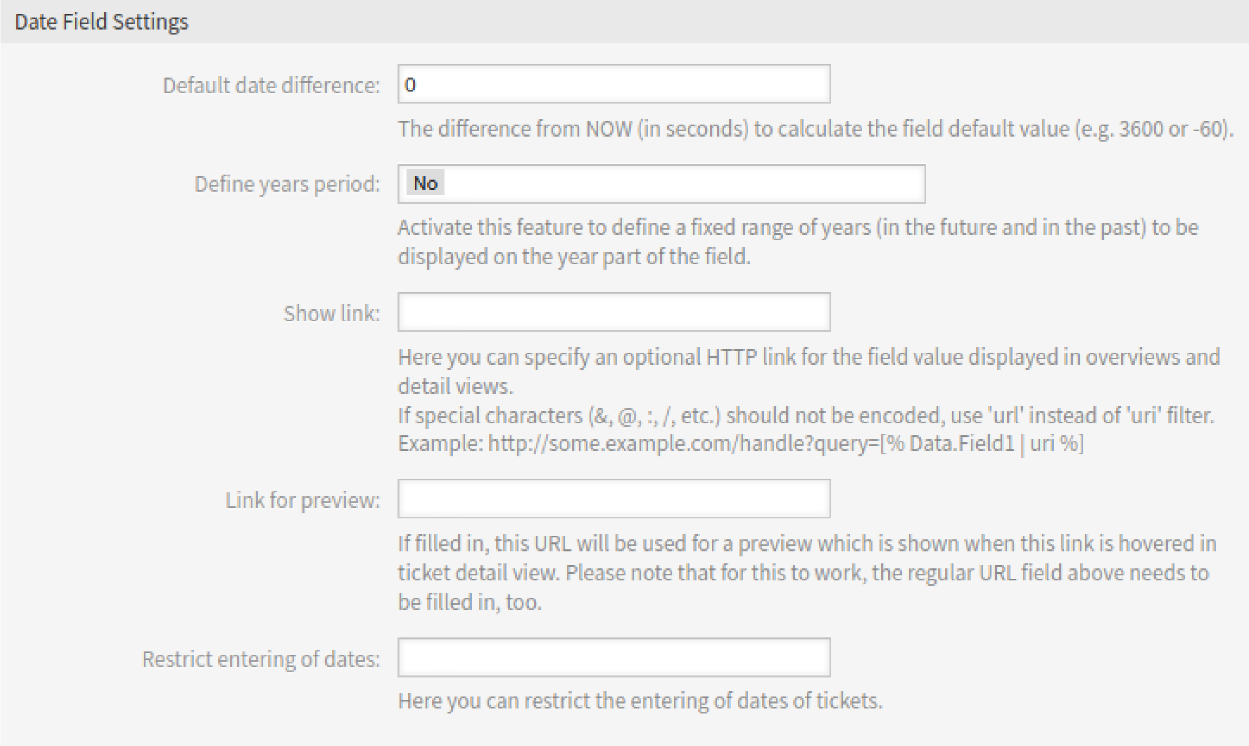
   See also:

   See *Display Dynamic Fields on Screens* for more information.
8. Go to *New Phone Ticket* screen, and notice that the new field is there. Add all needed information to the ticket.
9. Select an existing contact using autocomplete and choosing a contact.
10. Click on the *Create* button to create the ticket and go to the ticket detail view.
11. Open the widget configuration of the *Properties* widget in the ticket detail view, and enable the dynamic field in the *Hide/Show Properties* list.
12. The assigned contact and its attributes will be shown in the ticket detail view.
13. It is possible to update the attributes of the contact by clicking the *Edit contact* button that appears in the right side of dynamic field property card (if the current user is a member of the groups defined in system configuration setting `Frontend::Module###AdminDynamicFieldContactWithData`).

14. If 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.

Date Dynamic Field Settings

Date dynamic field is used to store a date value.



Date Field Settings

Default date difference: 0

The difference from NOW (in seconds) to calculate the field default value (e.g. 3600 or -60).

Define years period: No

Activate this feature to define a fixed range of years (in the future and in the past) to be displayed on the year part of the field.

Show link:

Here you can specify an optional HTTP link for the field value displayed in overviews and detail views.
If special characters (&, @, :, /, etc.) should not be encoded, use 'url' instead of 'uri' filter.
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 detail view. Please note that for this to work, the regular URL field above needs to be filled in, too.

Restrict entering of dates:

Here you can restrict the entering of dates of tickets.

Fig. 22: Date Dynamic Field Settings

Default date difference The difference from **now** (in seconds) to calculate the field default value (e.g. 3600 or -60).

Define years period Activate this feature to define a fixed range of years (in the future and in the past) to be displayed on the year part of the field. If set to *Yes* the following options will be available:

Years in the past Define the number of years in the past from the current day to display in the year selection for this dynamic field in edit screens.

Years in the future Define the number of years in the future from the current day to display in the year selection for this dynamic field in edit screens.

Note: If no years period is specified, the system uses the default value for the period: 10 years (5 years in the past and 5 years in the future).

Show link Here you can specify an optional HTTP link for the field value displayed in overviews and detail views. Example:

```
https://example.com/handle?query=[% Data.Field1 | uri %]
```

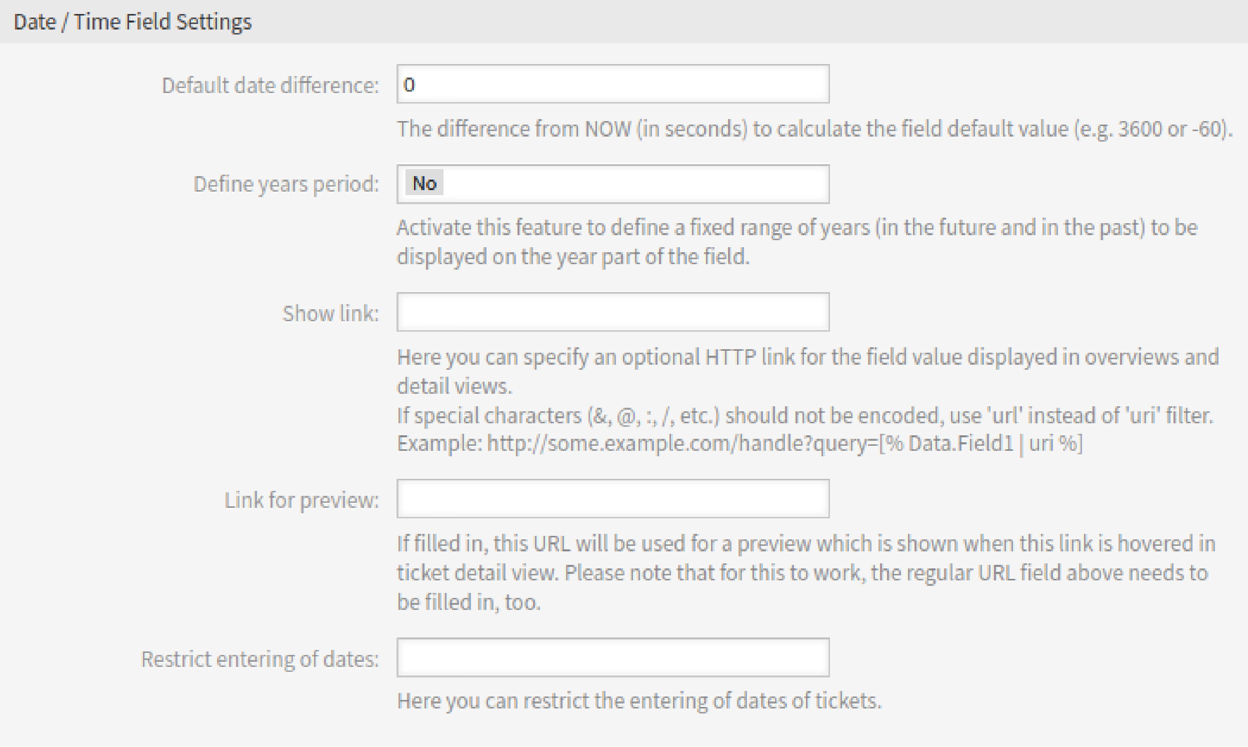
Link for preview If filled in, this URL will be used for a preview which is shown when this link is hovered in ticket detail view. Please note that for this to work, the regular URL field above needs to be filled in, too.

Restrict entering of dates Here you can restrict the entering of dates of tickets.

> **Prevent entry of dates in the future** Selecting this option will prevent entering a date that is after the current date.
>
> **Prevent entry of dates in the past** Selecting this option will prevent entering a date that is before the current date.

Date / Time Dynamic Field Settings

Date / time dynamic field is used to store a date time value.



Date / Time Field Settings

Default date difference: 0

The difference from NOW (in seconds) to calculate the field default value (e.g. 3600 or -60).

Define years period: No

Activate this feature to define a fixed range of years (in the future and in the past) to be displayed on the year part of the field.

Show link:

Here you can specify an optional HTTP link for the field value displayed in overviews and detail views.
If special characters (&, @, :, /, etc.) should not be encoded, use 'url' instead of 'uri' filter.
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 detail view. Please note that for this to work, the regular URL field above needs to be filled in, too.

Restrict entering of dates:

Here you can restrict the entering of dates of tickets.

Fig. 23: Date / Time Dynamic Field Settings

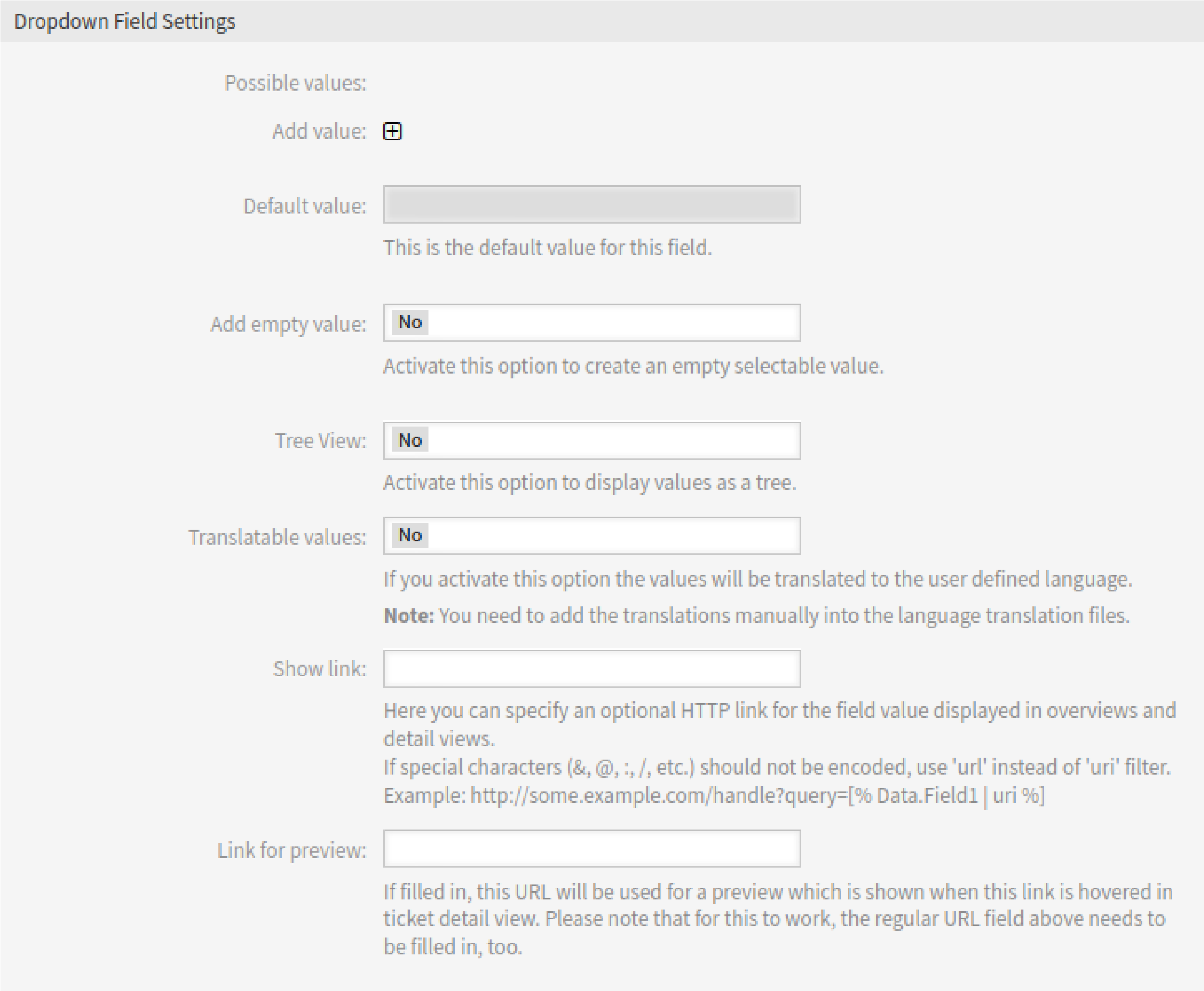
The settings for this type of dynamic field is the same as for date dynamic field.

Dropdown Dynamic Field Settings

Drop-down dynamic field is used to store a single value, from a closed list.

Possible values These are the possible data attributes for contacts. Clicking on the button will add two new fields, where a key (internal value) and a value (displayed value) can be set. With the button you can add multiple key-value pairs.

Default value This is the default value for this field and this will be shown on the edit screens.



Dropdown Field Settings

Possible values:

Add value: ⊞

Default value:

This is the default value for this field.

Add empty value: No

Activate this option to create an empty selectable value.

Tree View: No

Activate this option to display values as a tree.

Translatable values: No

If you activate this option the values will be translated to the user defined language.

Note: You need to add the translations manually into the language translation files.

Show link:

Here you can specify an optional HTTP link for the field value displayed in overviews and detail views.
If special characters (&, @, :, /, etc.) should not be encoded, use 'url' instead of 'uri' filter.
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 detail view. Please note that for this to work, the regular URL field above needs to be filled in, too.

Fig. 24: Dropdown Dynamic Field Settings

Add empty value If this option is activated an extra value is defined to show as a - in the list of possible values. This special value is empty internally.

Tree View Activate this option to display values as a tree.

Translatable values If you activate this option the values will be translated to the user defined language.

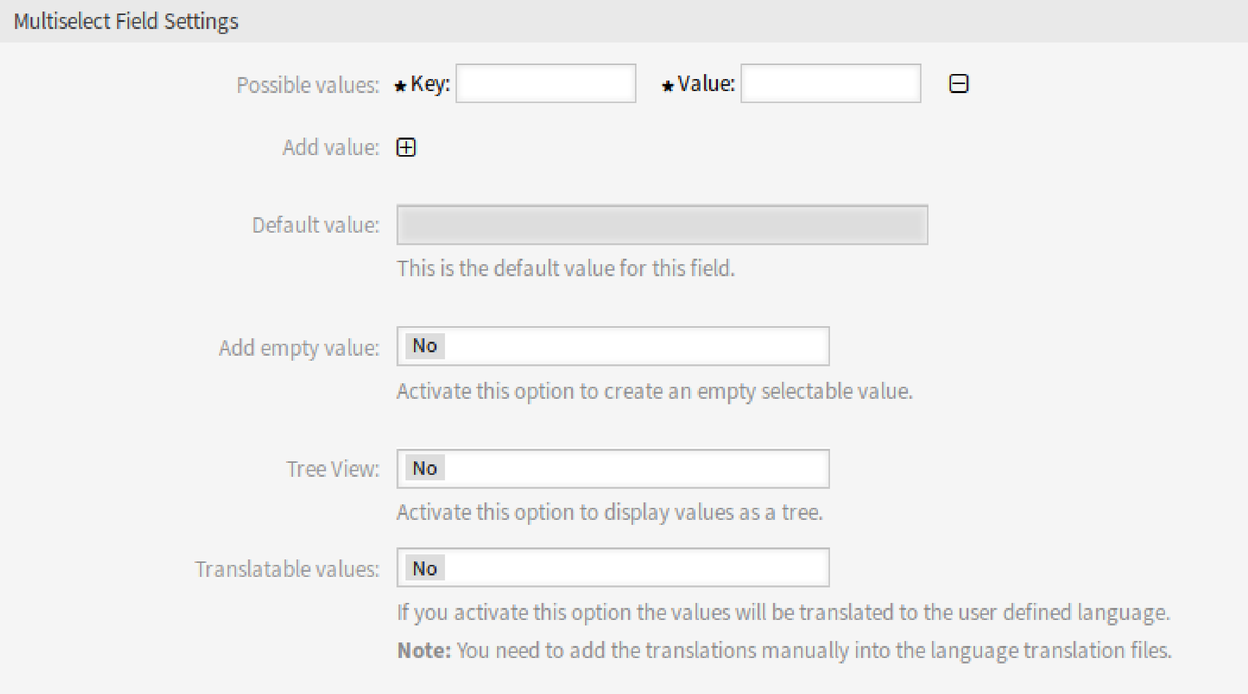
> **Note:** You need to add the translations manually into the language translation files.

Show link Here you can specify an optional HTTP link for the field value displayed in overviews and detail views. Example:

```
https://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 detail view. Please note that for this to work, the regular URL field above needs to be filled in, too.

Multiselect Dynamic Field Settings



Multiselect Field Settings

Possible values: ★Key: ★Value: ⊟

Add value: ⊞

Default value:

This is the default value for this field.

Add empty value: No

Activate this option to create an empty selectable value.

Tree View: No

Activate this option to display values as a tree.

Translatable values: No

If you activate this option the values will be translated to the user defined language.

Note: You need to add the translations manually into the language translation files.

Fig. 25: Multiselect Dynamic Field Settings

Possible values These are the possible data attributes for contacts. Clicking on the button will add two new fields, where a key (internal value) and a value (displayed value) can be set. With the button you can add multiple key-value pairs.

Default value This is the default value for this field and this will be shown on the edit screens.

Add empty value If this option is activated an extra value is defined to show as a - in the list of possible values. This special value is empty internally.

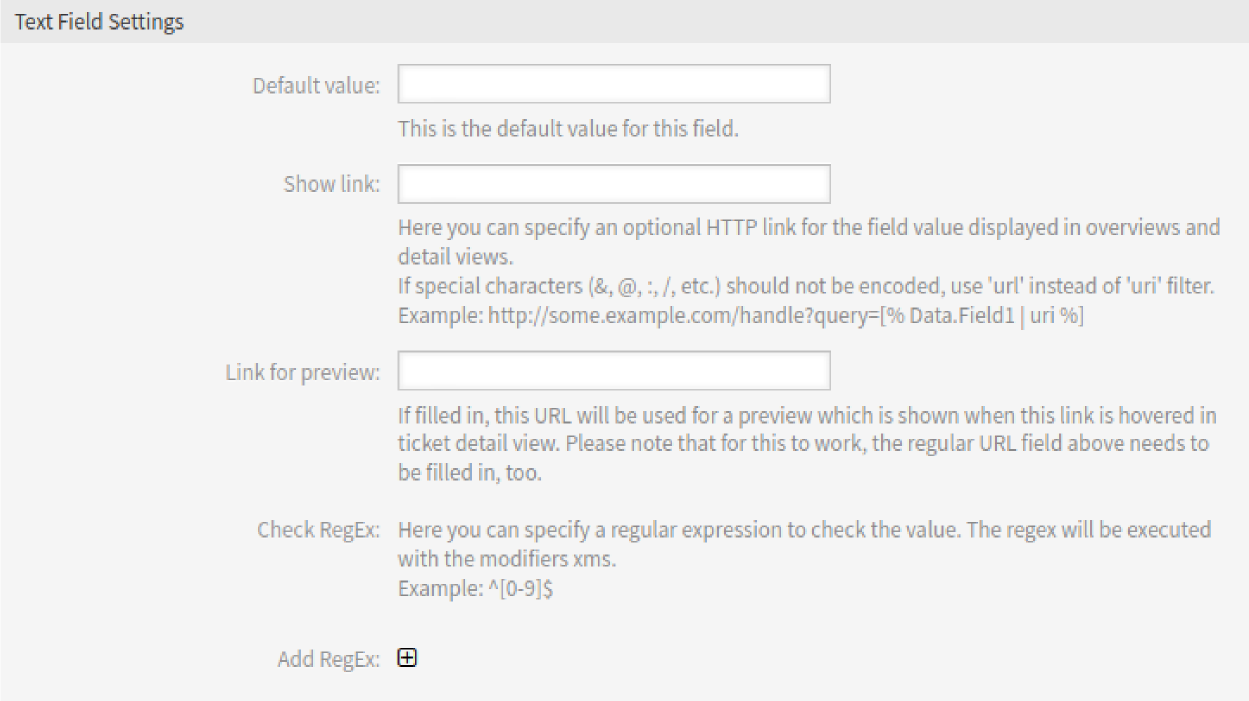
Tree View Activate this option to display values as a tree.

Translatable values If you activate this option the values will be translated to the user defined language.

> **Note:** You need to add the translations manually into the language translation files.

Text Dynamic Field Settings

Text dynamic field is used to store a single line string.



Text Field Settings

Default value:

This is the default value for this field.

Show link:

Here you can specify an optional HTTP link for the field value displayed in overviews and detail views.
If special characters (&, @, :, /, etc.) should not be encoded, use 'url' instead of 'uri' filter.
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 detail view. Please note that for this to work, the regular URL field above needs to be filled in, too.

Check RegEx: Here you can specify a regular expression to check the value. The regex will be executed with the modifiers xms.
Example: ^[0-9]$

Add RegEx: ⊞

Fig. 26: Text Dynamic Field Settings

Default value This is the default value for this field and this will be shown on the edit screens.

Show link Here you can specify an optional HTTP link for the field value displayed in overviews and detail views. Example:

```
https://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 detail view. Please note that for this to work, the regular URL field above needs to be filled in, too.

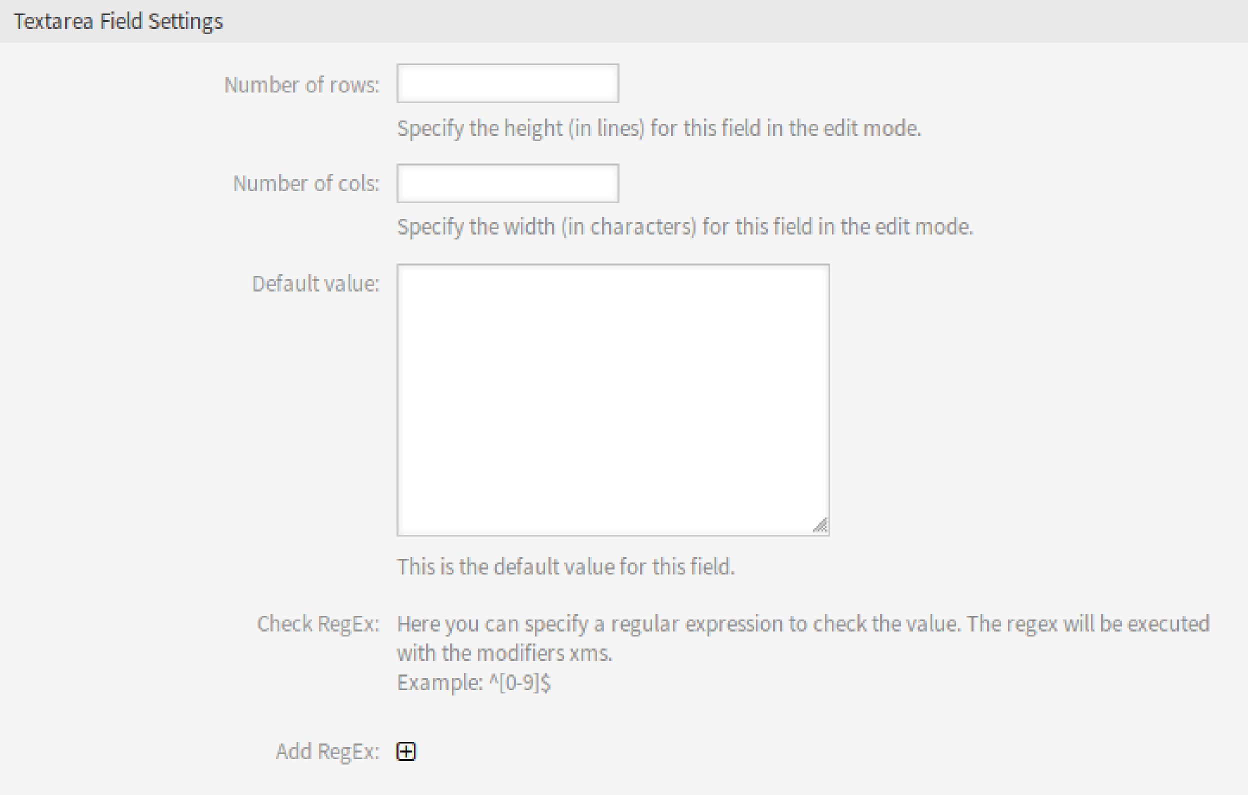
Check RegEx Here you can specify a regular expression to check the value. The regex will be executed with the modifiers `xms`. Example:

```
^[0-9]$
```

Add RegEx Clicking on the button will add two new fields, where a regular expression and an error message can be added.

Textarea Dynamic Field Settings

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



Textarea Field Settings

Number of rows:

Specify the height (in lines) for this field in the edit mode.

Number of cols:

Specify the width (in characters) for this field in the edit mode.

Default value:

This is the default value for this field.

Check RegEx: Here you can specify a regular expression to check the value. The regex will be executed with the modifiers xms.
Example: ^[0-9]$

Add RegEx: ⊞

Fig. 27: Textarea Dynamic Field Settings

Number of rows The height (in lines) for this field in the edit mode.

Number of cols The width (in characters) for this field in the edit mode.

Default value This is the default value for this field and this will be shown on the edit screens.

Check RegEx Here you can specify a regular expression to check the value. The regex will be executed with the modifiers `xms`. Example:

```
^[0-9]$
```

Add RegEx Clicking on the button will add two new fields, where a regular expression and an error message can be added.

Database Dynamic Field

Note: To use this type of dynamic field, an **OTRS** service package is needed. Please contact at sales@otrs.com for an upgrade.

Web Service Dynamic Field

Note: To use this type of dynamic field, an **OTRS** service package is needed. Please contact at sales@otrs.com for an upgrade.

5.2.3 Display Dynamic Fields on Screens

The following sections describe how to add a dynamic field to certain screens. The following examples use a dynamic field named `Test1`. Please make sure that you replace it with the actual name of your dynamic field.

Note: Make sure that the *Validity* of the dynamic field is set to *valid*.

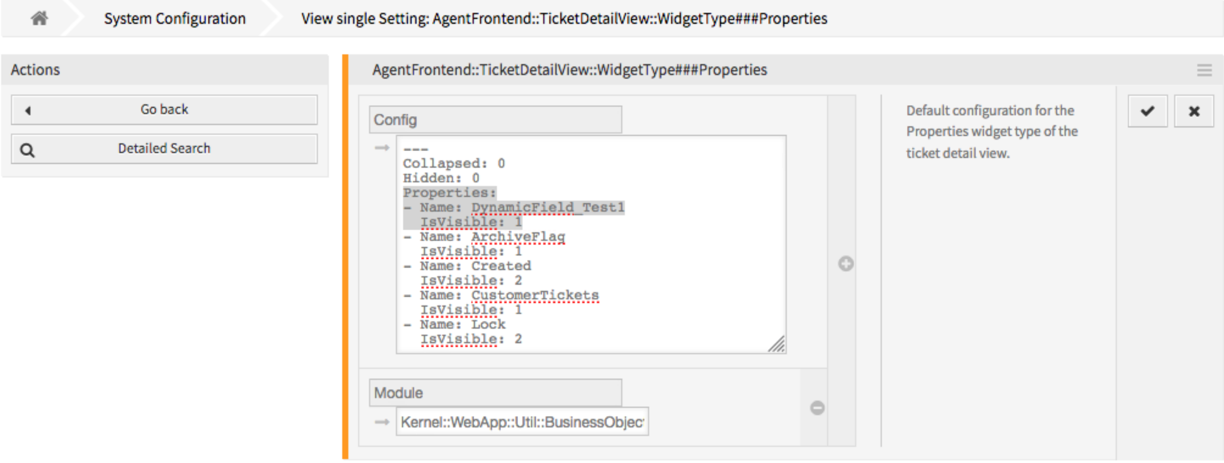
Dynamic Field in Business Object Detail View

Dynamic fields can be added to any widget of the business object detail view. The following examples show the possibilities.

To add a **ticket** dynamic field to the *Properties* widget:

1. Go to the *System Configuration* screen.
2. Search for the setting `AgentFrontend::TicketDetailView::WidgetType###Properties`.
3. Add the ticket dynamic field to the `Properties` section of the YAML configuration.

```
Properties:
- Name: DynamicField_Test1
  IsVisible: 1
```



System Configuration

View single Setting: AgentFrontend::TicketDetailView::WidgetType###Properties

Actions

Go back

Detailed Search

AgentFrontend::TicketDetailView::WidgetType###Properties

Config

```
---
Collapsed: 0
Hidden: 0
Properties:
- Name: DynamicField_Test1
  IsVisible: 1
- Name: ArchiveFlag
  IsVisible: 1
- Name: Created
  IsVisible: 2
- Name: CustomerTickets
  IsVisible: 1
- Name: Lock
  IsVisible: 2
```

Module

Kernel::WebApp::Util::BusinessObjec

Default configuration for the Properties widget type of the ticket detail view.

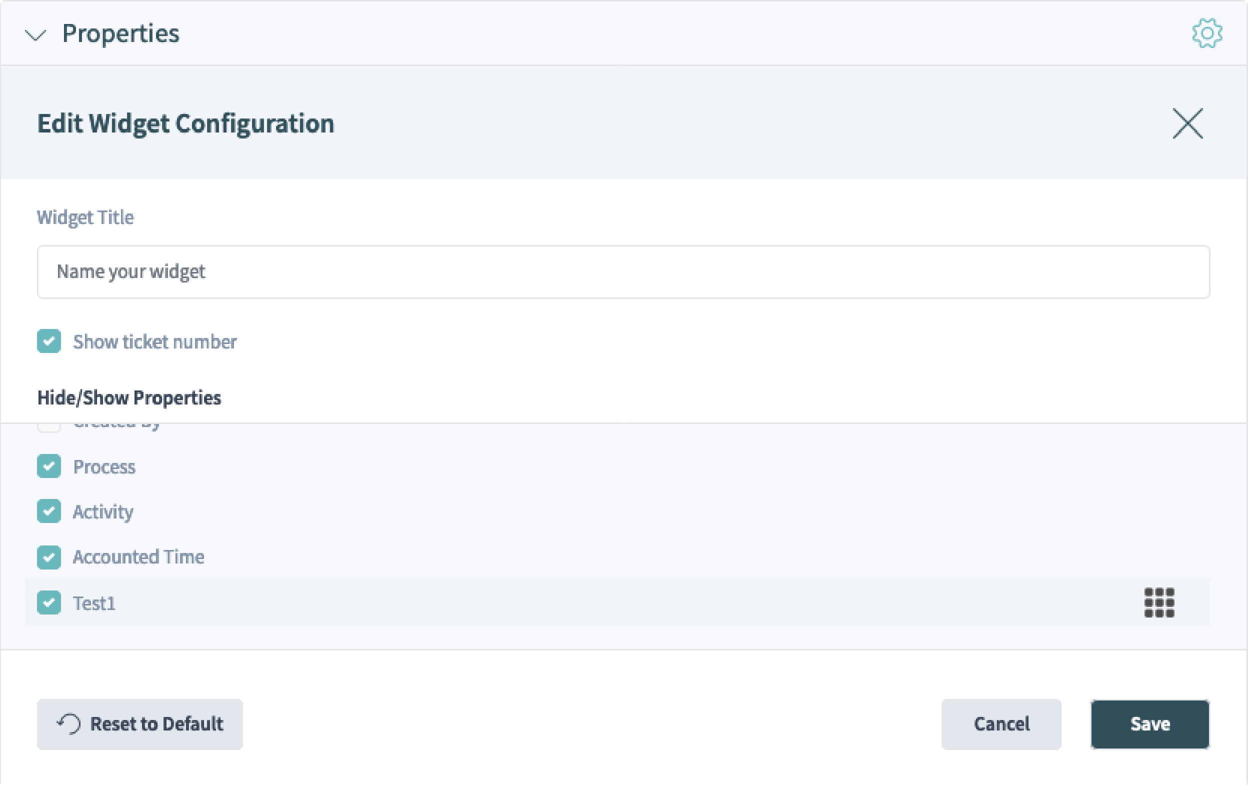
Fig. 28: Add Dynamic Field to Properties Widget Configuration

See also:

Detailed information about the configuration of the property card and possible keys can be found under *the reference*.

4. Deploy the modified system configuration.

5. Go to the ticket detail view screen in the agent interface.
6. Find the *Properties* widget and open its configuration via the gear icon. Identify the `Test1` dynamic field property in the *Hide/Show Properties* list. Activate it by clicking on the checkbox. Click on the *Save* button to apply the change and close the widget configuration.



Widget Title

Name your widget

☑ Show ticket number

Hide/Show Properties

□ Created By

☑ Process

☑ Activity

☑ Accounted Time

☑ Test1

Fig. 29: Enable Dynamic Field Property Card in Properties Widget

If defined, the ticket dynamic field value will then be displayed as a separate property card.

To add an **article** dynamic field to the *Communication Stream* widget:

1. Go to the *System Configuration* screen.
2. Search for the setting `AgentFrontend::TicketDetailView::WidgetType###CommunicationStream`.
3. Add the article dynamic field name to the `ArticleDynamicFields` array of the YAML configuration.

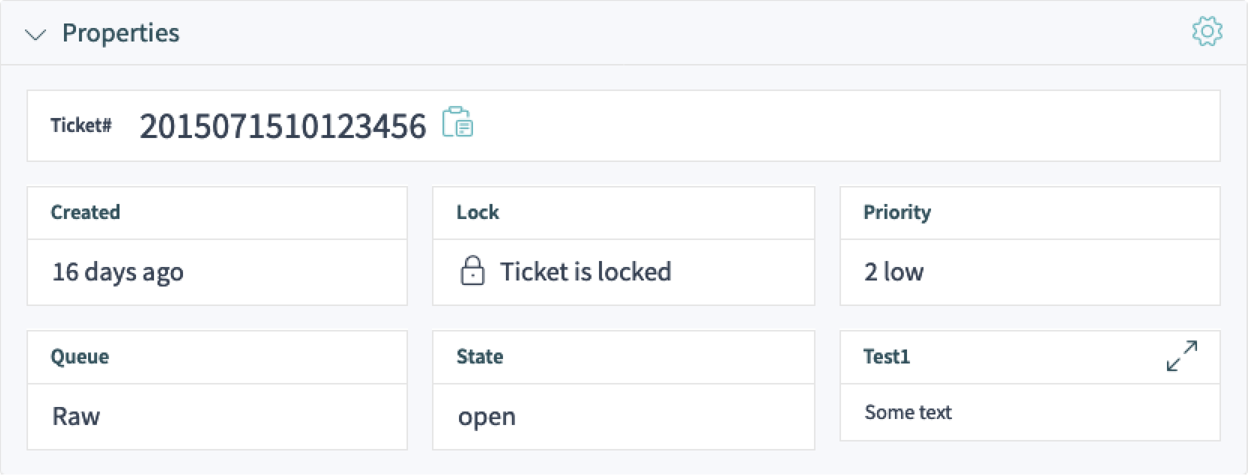
```
ArticleDynamicFields:
- Test1
```

4. Deploy the modified system configuration.

If defined, the article dynamic field value will then be displayed as a separate line in the article header in the *Communication Stream* widget. If not visible, make sure the header is expanded by clicking on it.

To add a **ticket** dynamic field to the *Business Process Information* widget:

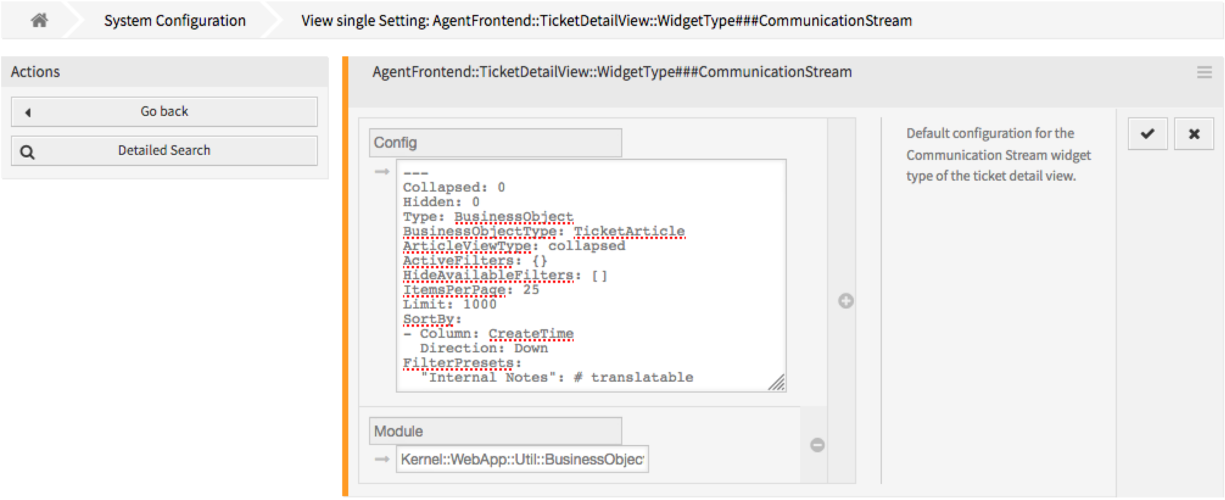
1. Go to the *System Configuration* screen.
2. Search for the setting `AgentFrontend::TicketDetailView::Widget::BusinessProcessInformation###D`
3. Add the dynamic field to the list and enabled it.



Properties

Ticket# 2015071510123456		
Created	Lock	Priority
16 days ago	Ticket is locked	2 low
Queue	State	Test1
Raw	open	Some text

Fig. 30: Display Dynamic Field Property Card in Properties Widget



System Configuration
View single Setting: AgentFrontend::TicketDetailView::WidgetType###CommunicationStream
Actions
Go back
Detailed Search

AgentFrontend::TicketDetailView::WidgetType###CommunicationStream

Config

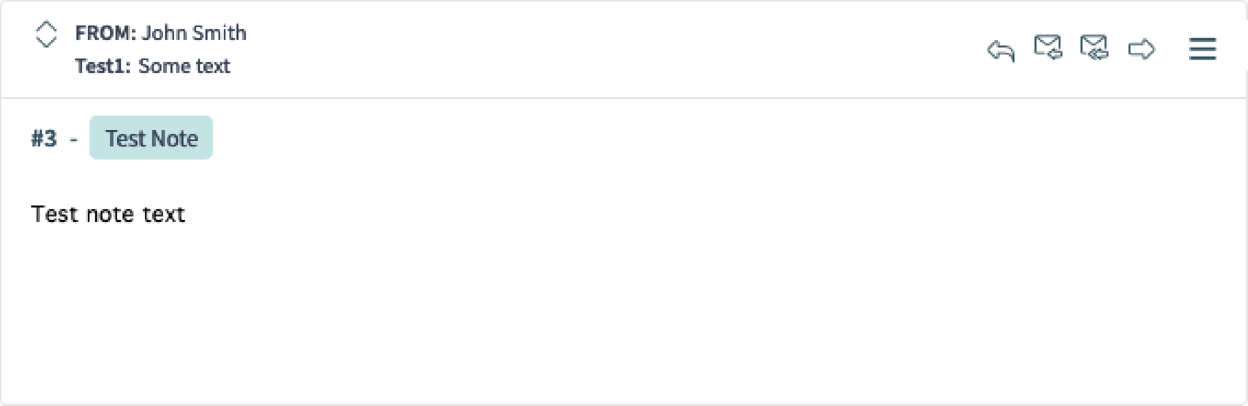
```
---
Collapsed: 0
Hidden: 0
Type: BusinessObject
BusinessObjectType: TicketArticle
ArticleViewType: collapsed
ActiveFilters: {}
HideAvailableFilters: []
ItemsPerPage: 25
Limit: 1000
SortBy:
- Column: CreateTime
  Direction: Down
FilterPresets:
  "Internal Notes": # translatable
```

Module

Kernel::WebApp::Util::BusinessObjec

Default configuration for the Communication Stream widget type of the ticket detail view.

Fig. 31: Add Dynamic Field to Communication Stream Widget Configuration

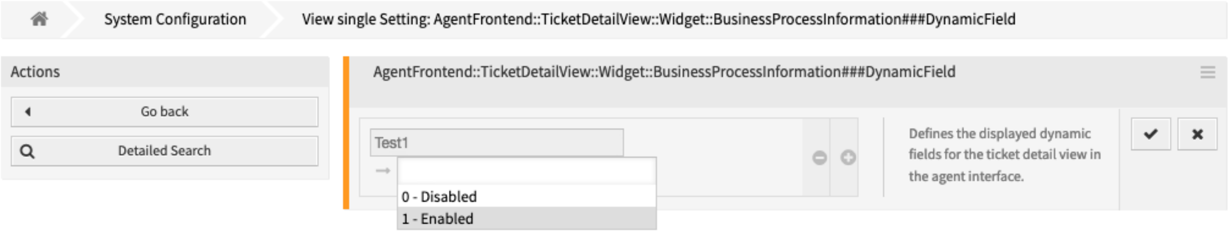


FROM: John Smith
Test1: Some text

#3 - Test Note

Test note text

Fig. 32: Display Dynamic Field Value in Communication Stream Widget



System Configuration

View single Setting: AgentFrontend::TicketDetailView::Widget::BusinessProcessInformation###DynamicField

Actions

Go back

Detailed Search

AgentFrontend::TicketDetailView::Widget::BusinessProcessInformation###DynamicField

Test1

0 - Disabled

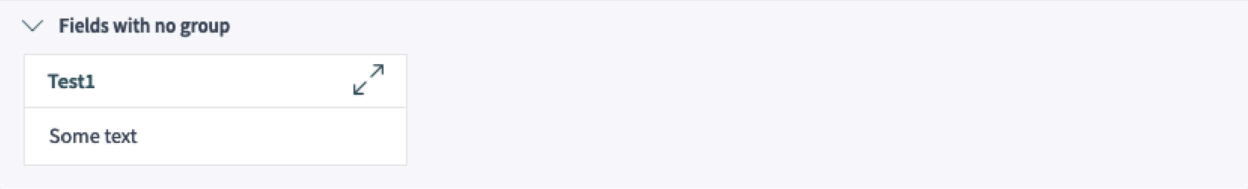
1 - Enabled

Defines the displayed dynamic fields for the ticket detail view in the agent interface.

Fig. 33: Add Dynamic Field to Business Process Information Widget Configuration

4. Deploy the modified system configuration.

If defined, the ticket dynamic field value will then be displayed as a separate property card in the *Business Process Information* widget of a process ticket.



Fields with no group

Test1

Some text

Fig. 34: Display Dynamic Field Property Card in Business Process Information Widget

See also:

If you want to group the dynamic fields into groups, please check the setting AgentFrontend::TicketDetailView::Widget::BusinessProcessInformation###DynamicFieldGroup.

Dynamic Field in Ticket and Article Action

The following example shows how to add a dynamic field to the *Close Ticket* action. The steps are identical for other actions, only the system configuration keys are different for each form.

See also:

For the names and explanations of other forms, please consult the *Form Fields* chapter.

To add a **ticket** or an **article** dynamic field to the *Close Ticket* form:

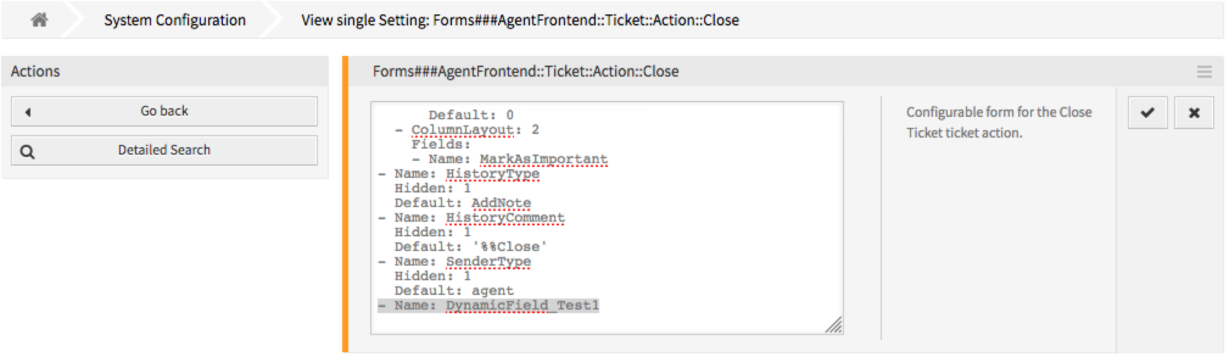
1. Go to the *System Configuration* screen.
2. Search for the setting `Forms###AgentFrontend::Ticket::Action::Close`.
3. Add the dynamic field to the `Fields` section of the YAML configuration.

```
Fields:
- Name: DynamicField_Test1
```

4. Deploy the modified system configuration.

The ticket dynamic field will be displayed in the relevant ticket or article action as a form field.

Note: Article dynamic fields are only displayed if article related fields such as `Subject` and `Body` are present. If these fields do not exist, no article is created and no value is set for the dynamic field when the



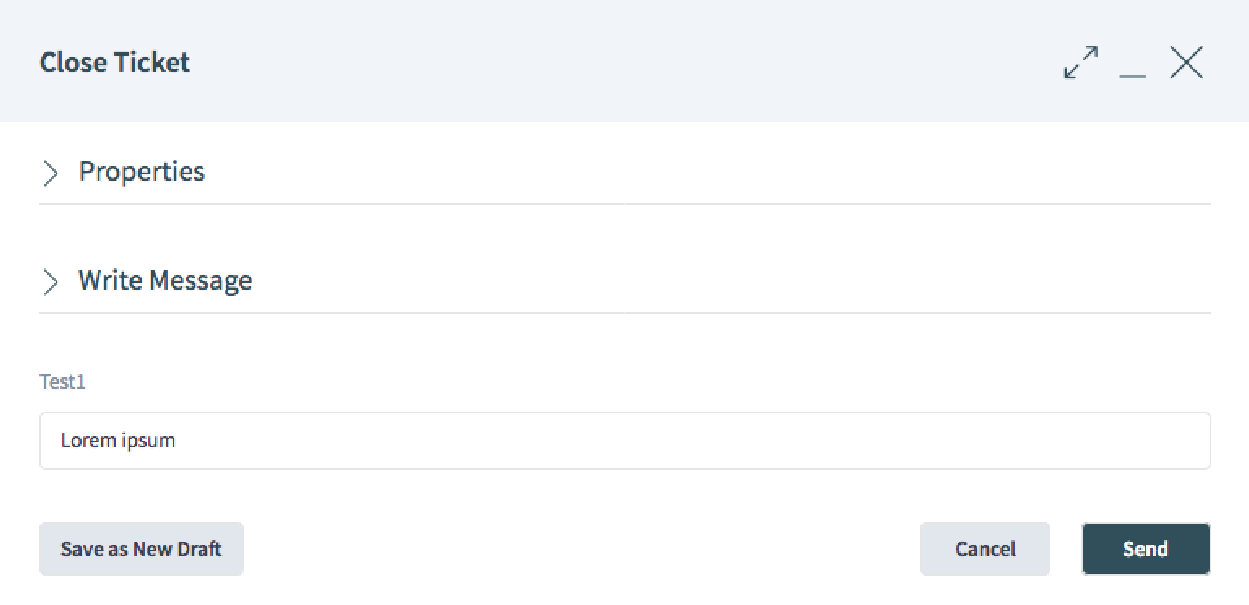
System Configuration
View single Setting: Forms###AgentFrontend::Ticket::Action::Close
Actions
Go back
Detailed Search

Forms###AgentFrontend::Ticket::Action::Close

```
      Default: 0
  - ColumnLayout: 2
    Fields:
    - Name: MarkAsImportant
- Name: HistoryType
  Hidden: 1
  Default: AddNote
- Name: HistoryComment
  Hidden: 1
  Default: '%%Close'
- Name: SenderType
  Hidden: 1
  Default: agent
- Name: DynamicField_Test1
```

Configurable form for the Close Ticket ticket action.

Fig. 35: Add Dynamic Field to Close Ticket Form Configuration



Test1

Lorem ipsum

Fig. 36: Display Dynamic Field in Close Ticket Action

action is executed.

Dynamic Field in Business Object List

The following examples show how to add a dynamic field as a column or as a filter to the organizer item type *Ticket List*. This affects all organizer items of the type *Ticket List*. If only a specific organizer item needs to be affected, then please modify the system configuration key of the relevant organizer item.

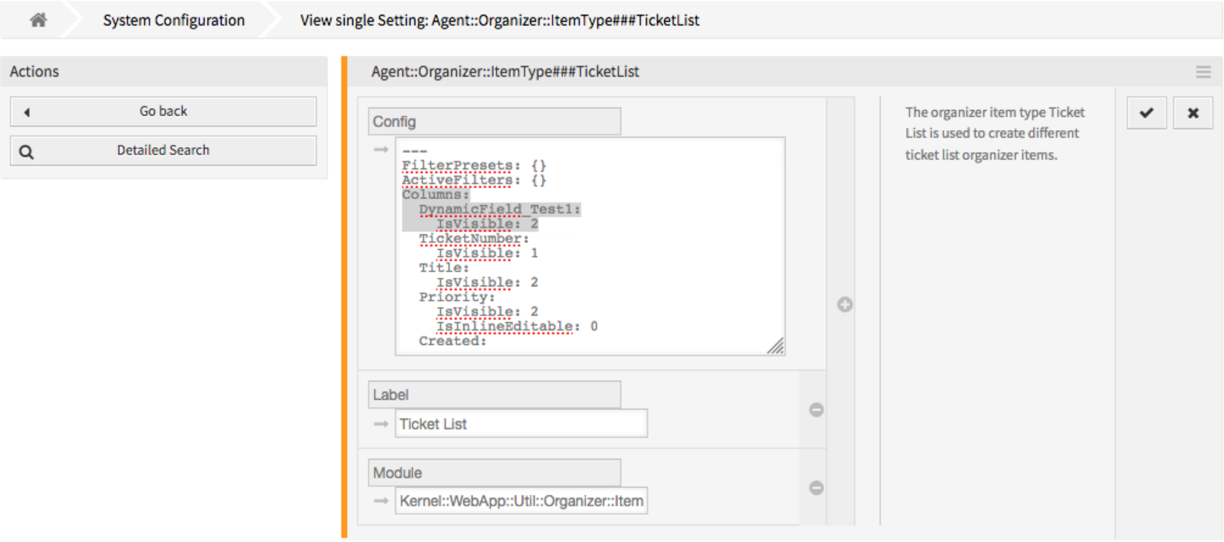
See also:

For the names and explanations for other organizer items please read the chapter *Business Object Lists*.

To add a **ticket** dynamic field as a list column:

1. Go to the *System Configuration* screen.
2. Search for the setting `Agent::Organizer::ItemType###TicketList`.
3. Add the ticket dynamic field to the `Columns` section of the YAML configuration.

```
Columns:
  DynamicField_Test1:
    IsVisible: 2
```



System Configuration
View single Setting: Agent::Organizer::ItemType###TicketList
Actions
Go back
Detailed Search
Agent::Organizer::ItemType###TicketList
Config
Label
Module

```
---
FilterPresets: {}
ActiveFilters: {}
Columns:
  DynamicField_Test1:
    IsVisible: 2
  TicketNumber:
    IsVisible: 1
  Title:
    IsVisible: 2
  Priority:
    IsVisible: 2
    IsInlineEditable: 0
  Created:
```

Label: Ticket List

Module: Kernel::WebApp::Util::Organizer::Item

The organizer item type Ticket List is used to create different ticket list organizer items.

Fig. 37: Add Dynamic Field to Ticket List Configuration

See also:

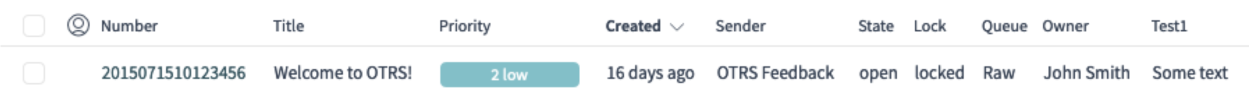
For detailed information about the column configuration and possible keys, please check *the reference*.

4. Deploy the modified system configuration.

If defined, the ticket dynamic field value will then be displayed in a column cell, for any of the organizer items of the ticket list type.

To add a **ticket** dynamic field as a list filter:

1. Go to the *System Configuration* screen.

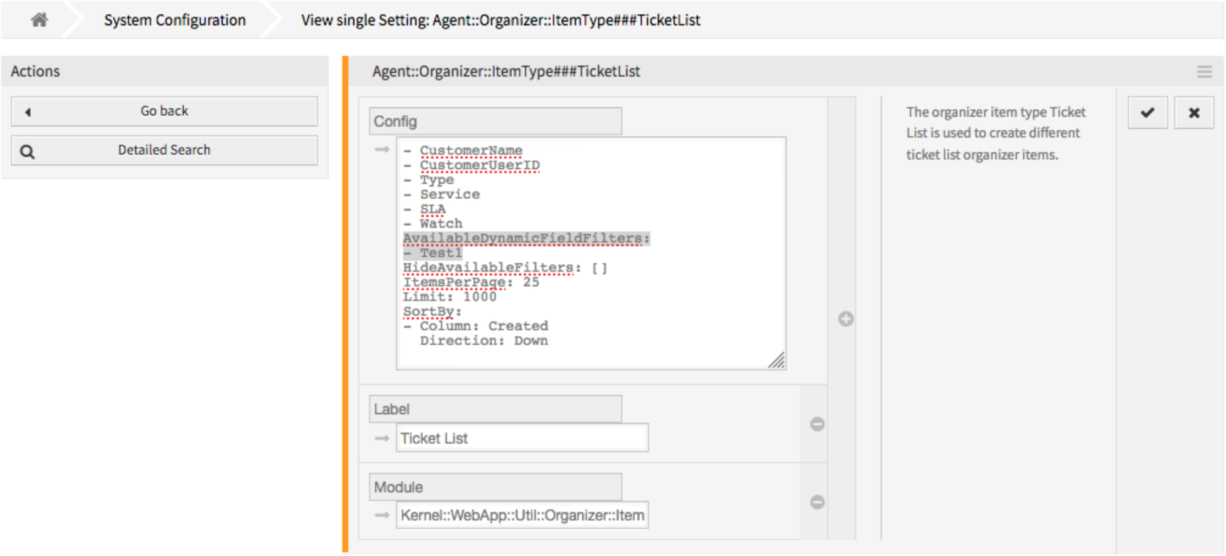


	Number	Title	Priority	Created	Sender	State	Lock	Queue	Owner	Test1
□	2015071510123456	Welcome to OTRS!	2 low	16 days ago	OTRS Feedback	open	locked	Raw	John Smith	Some text

Fig. 38: Display Dynamic Field Value in Ticket List

2. Search for the setting `Agent::Organizer::ItemType###TicketList`.
3. Add the ticket dynamic field name to the `AvailableDynamicFieldFilters` array of the YAML configuration.

```
AvailableDynamicFieldFilters:
- Test1
```



System Configuration
View single Setting: Agent::Organizer::ItemType###TicketList
Actions
Go back
Detailed Search
Agent::Organizer::ItemType###TicketList
Config

```
- CustomerName
- CustomerUserID
- Type
- Service
- SLA
- Watch
AvailableDynamicFieldFilters:
- Test1
HideAvailableFilters: []
ItemsPerPage: 25
Limit: 1000
SortBy:
- Column: Created
  Direction: Down
```

Ticket List

Kernel::WebApp::Util::Organizer::Item

The organizer item type Ticket List is used to create different ticket list organizer items.

Fig. 39: Add Dynamic Field Filter to Ticket List Configuration

4. Deploy the modified system configuration.

The ticket dynamic field can now be used as a filter in the business object list.

Dynamic Field in Ticket Create Screen

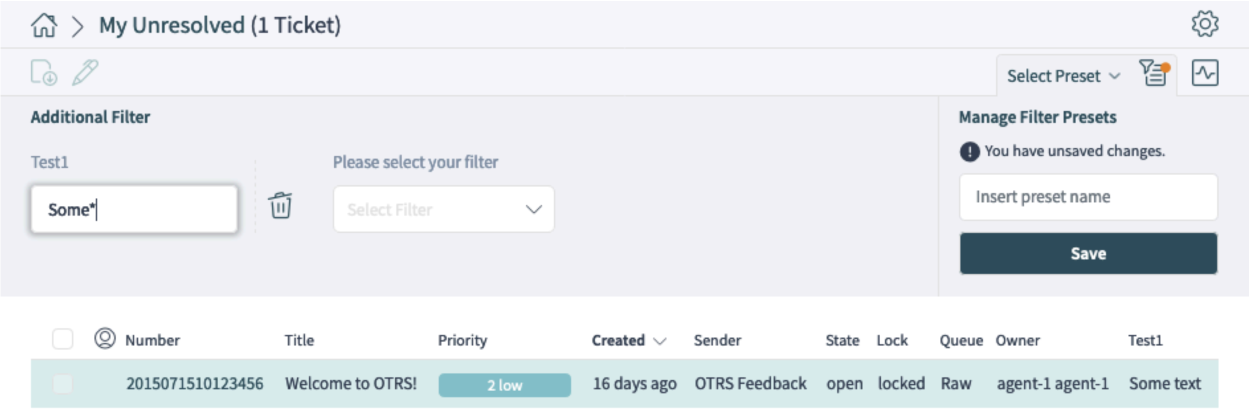
To add a **ticket** or an **article** dynamic field to *New Email Ticket* screen:

1. Go to the *System Configuration* screen.
2. Search for the setting `Forms###AgentFrontend::TicketCreate::Email::CreateProperties`.
3. Add the ticket dynamic field to the `Fields` section of the YAML configuration.

```
Fields:
- Name: DynamicField_Test1
```

See also:

For more information about form fields and groups, please consult the *Forms* chapter.



My Unresolved (1 Ticket)

Select Preset

Additional Filter

Test1: Some*

Please select your filter: Select Filter

Manage Filter Presets

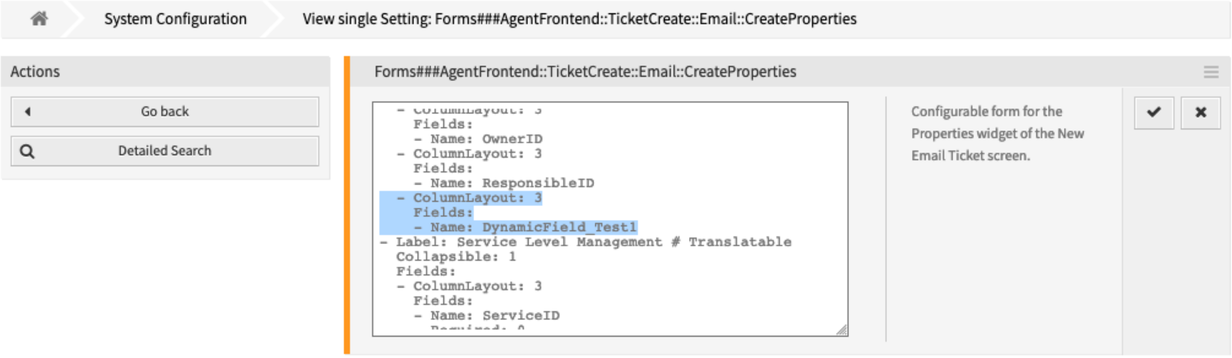
You have unsaved changes.

Insert preset name

Save

	Number	Title	Priority	Created	Sender	State	Lock	Queue	Owner	Test1
□	2015071510123456	Welcome to OTRS!	2 low	16 days ago	OTRS Feedback	open	locked	Raw	agent-1 agent-1	Some text

Fig. 40: Use Dynamic Field as Filter in Ticket List



System Configuration > View single Setting: Forms###AgentFrontend::TicketCreate::Email::CreateProperties

Actions

Go back

Detailed Search

Forms###AgentFrontend::TicketCreate::Email::CreateProperties

```
  - ColumnLayout: 3
    Fields:
    - Name: OwnerID
  - ColumnLayout: 3
    Fields:
    - Name: ResponsibleID
  - ColumnLayout: 3
    Fields:
    - Name: DynamicField_Test1
- Label: Service Level Management # Translatable
  Collapsible: 1
  Fields:
  - ColumnLayout: 3
    Fields:
    - Name: ServiceID
      Required: 0
```

Configurable form for the Properties widget of the New Email Ticket screen.

Fig. 41: Add Dynamic Field to New Email Ticket Configuration

4. Deploy the modified system configuration.

The ticket or article dynamic field can now be used as part of the *New Email Ticket* screen.

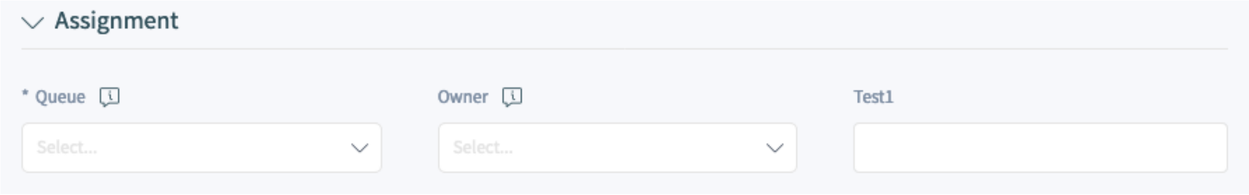
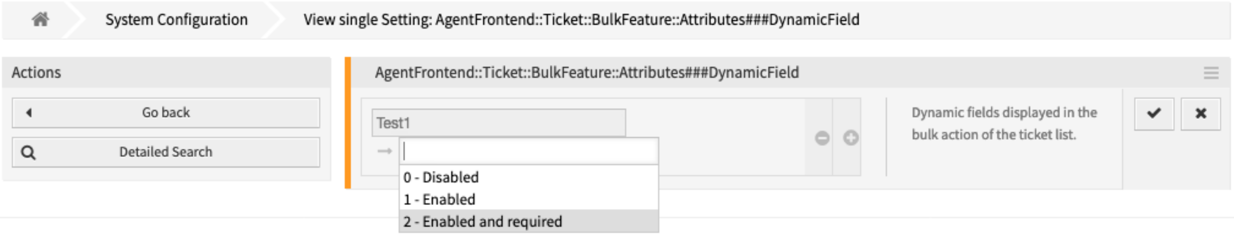


Fig. 42: Display Dynamic Field in New Email Ticket Screen

Dynamic Field in Ticket Bulk Action

To add a **ticket** or an **article** dynamic field to a ticket bulk action:

1. Go to the *System Configuration* screen.
2. Search for the setting `AgentFrontend::Ticket::BulkFeature::Attributes###DynamicField`.
3. Add the dynamic field to the list and enabled it.



System Configuration
View single Setting: AgentFrontend::Ticket::BulkFeature::Attributes###DynamicField
Actions
Go back
Detailed Search
AgentFrontend::Ticket::BulkFeature::Attributes###DynamicField
Test1
0 - Disabled
1 - Enabled
2 - Enabled and required
Dynamic fields displayed in the bulk action of the ticket list.

Fig. 43: Add Dynamic Field to Ticket Bulk Configuration

4. Deploy the modified system configuration.

The ticket or article dynamic field is now displayed in the *Properties* section in the action form of the bulk action.

Dynamic Field in the Print Output

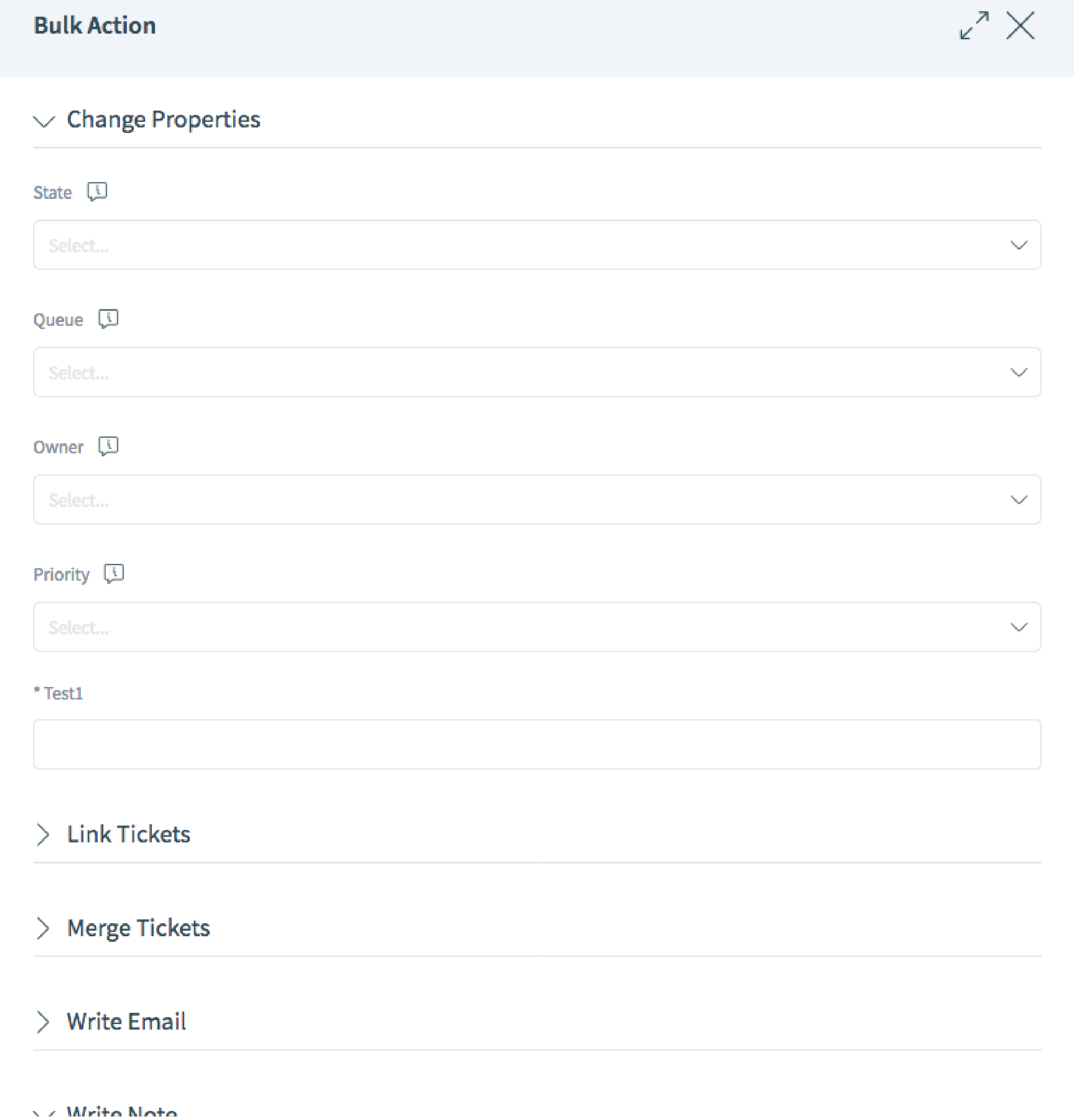
To add a **ticket** or an **article** dynamic field to the print output:

1. Go to the *System Configuration* screen.
2. Search for the setting `AgentFrontend::Ticket::Print###DynamicField`.
3. Add the dynamic field to the list and enabled it.
4. Deploy the modified system configuration.

The ticket or article dynamic field is now displayed in the printed output of the ticket.

Dynamic Field in External Interface

To add a dynamic field to the external interface:



Bulk Action

Change Properties

State

Select...

Queue

Select...

Owner

Select...

Priority

Select...

* Test1

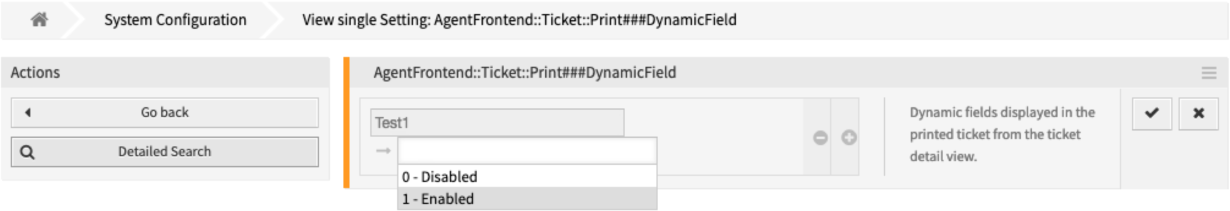
Link Tickets

Merge Tickets

Write Email

Write Note

Fig. 44: Display Dynamic Field in Ticket Bulk Action



System Configuration | View single Setting: AgentFrontend::Ticket::Print###DynamicField

Actions

Go back

Detailed Search

AgentFrontend::Ticket::Print###DynamicField

Test1

0 - Disabled

1 - Enabled

Dynamic fields displayed in the printed ticket from the ticket detail view.

Fig. 45: Add Dynamic Field to Print Ticket Configuration

Ticket Dynamic Fields

Test1: Some text

Fig. 46: Display Dynamic Field in Printed Output

1. Go to *System Configuration* screen.
2. Navigate to *Frontend* → *External* → *View* in the navigation tree.
3. Select the screen where the dynamic field should be displayed.
4. Search for the dynamic field setting of the screen and click on the *Edit this setting* button.

   The following settings are relevant to adding dynamic fields:

   ```
   - ExternalFrontend::KnowledgeBaseDetailView###DynamicField
   - ExternalFrontend::TicketCreate###DynamicField
   - ExternalFrontend::TicketDetailView###DynamicField
   - ExternalFrontend::TicketDetailView###FollowUpDynamicField
   - ExternalFrontend::TicketOverview###DynamicField
   ```

5. Click on the + button to add the dynamic field. The key is the name of the dynamic field, the value is *1 - Enabled*. This setting is used to display the content of dynamic field in the selcted screen.
6. Search for setting `ExternalFrontend::TicketDetailView###FollowUpDynamicField` and click on the *Edit this setting* button.
7. Click on the + button to add the dynamic field. The key is the name of the dynamic field, the value is *1 - Enabled*. This setting is used to configure dynamic fields in the answer part of customers detail view.
8. Repeat the steps for other views, if needed.
9. Deploy the modified settings.

5.2.4 Add Dynamic Fields to Search Engine

By default, the content of the dynamic fields cannot be searched using the document search functionality. Each dynamic field has to be added manually.

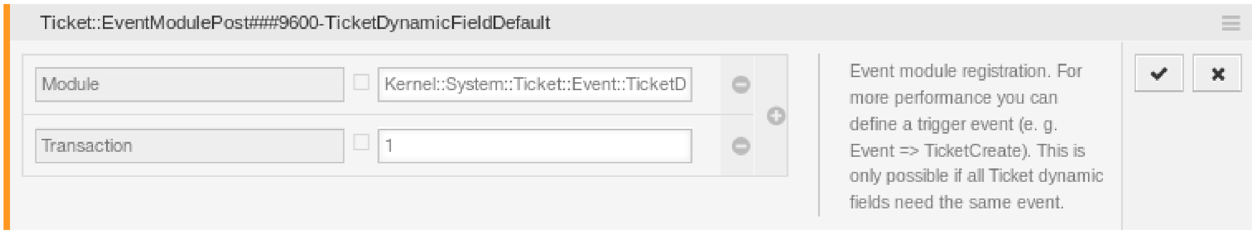
To add a dynamic field to the document search functionality:

1. Go to *System Configuration* screen.
2. Navigate to *Frontend* → *Agent* → *DocumentSearch* and *Frontend* → *External* → *DocumentSearch* in the navigation tree.
3. Search for the setting `DocumentSearch::Agent::DynamicField` and `DocumentSearch::External::DynamicField` respectively.
4. Click on the *Edit this setting* button.
5. Click on the + button to add the dynamic field.
6. Enter the name of the dynamic field to the text box and click on the tick button.
7. Select *0 - Disabled* or *1 - Enabled*.
8. Click on the tick button on the right to save the setting.
9. Deploy the modified system configuration.

5.2.5 Set Default Value via Ticket Event Module

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

1. Go to the *System Configuration* screen.
2. Navigate to *Core* → *Event* → *Ticket* and search for the setting `Ticket::EventModulePost###9600-TicketDynamicFieldDefault`.
3. Click on the *Edit this setting* button to activate the setting.
4. Click on the tick button on the right to save the setting.
5. Deploy the modified system configuration.



Ticket::EventModulePost###9600-TicketDynamicFieldDefault

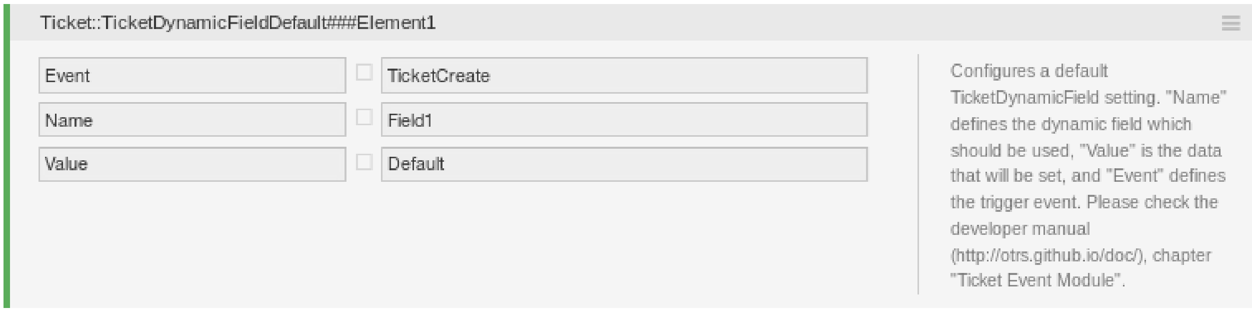
Module	□	Kernel::System::Ticket::Event::TicketD
Transaction	□	1

Event module registration. For more performance you can define a trigger event (e. g. Event => TicketCreate). This is only possible if all Ticket dynamic fields need the same event.

Fig. 47: Activate Ticket Event Module

Example: activate *Field1* in `TicketCreate` event:

1. Go to the *System Configuration* screen.
2. Navigate to *Core* → *Ticket* → *DynamicFieldDefault* and search for the setting `Ticket::TicketDynamicFieldDefault###Element1`.
3. Click on the *Edit this setting* button to activate the setting.
4. Click on the tick button on the right to save the setting.
5. Deploy the modified system configuration.



Ticket::TicketDynamicFieldDefault###Element1

Event	□	TicketCreate
Name	□	Field1
Value	□	Default

Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://otrs.github.io/doc/), chapter "Ticket Event Module".

Fig. 48: Activate Dynamic Field in Ticket Create Event

Note: This configuration can be set in any of the 16 `Ticket::TicketDynamicFieldDefault###Element` settings.

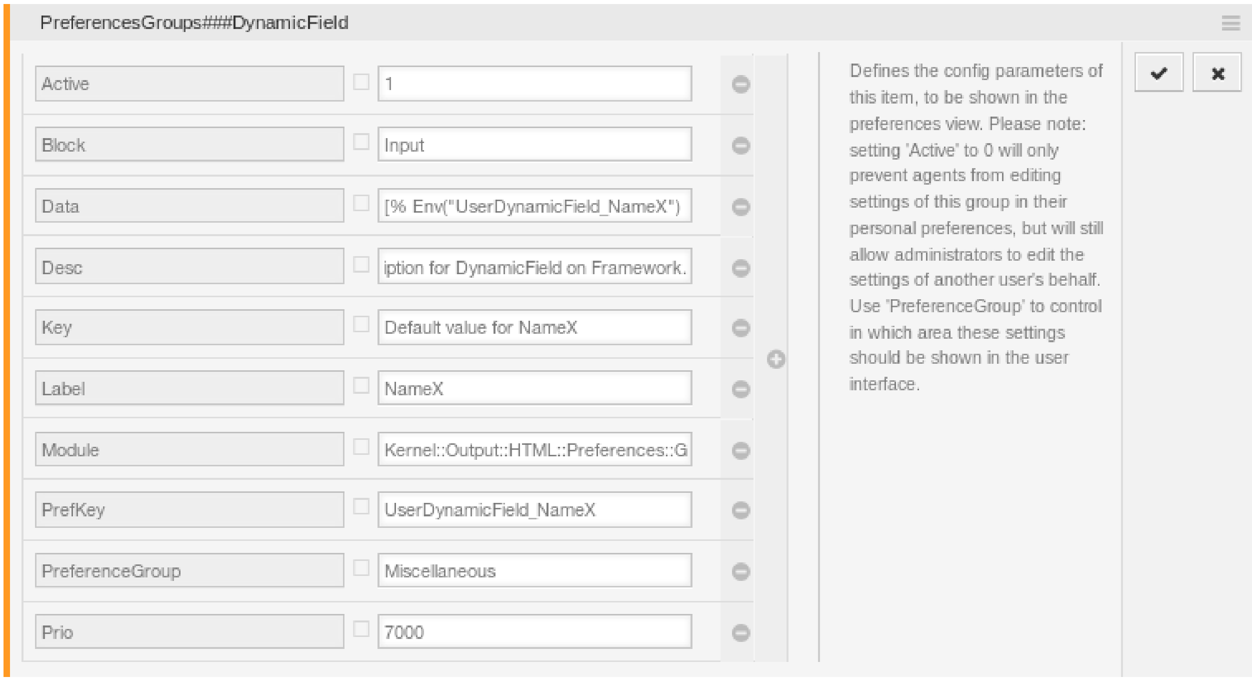
See also:

If more than 16 fields needs to be set up, a custom XML file must be placed in `$OTRS_HOME/Kernel/Config/Files/XML` directory to extend this feature.

5.2.6 Set Default Value via User Preferences

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

1. Go to the *System Configuration* screen.
2. Navigate to *Frontend* → *Agent* → *View* → *Preferences* and search for the setting `PreferencesGroups###DynamicField`.
3. Click on the *Edit this setting* button to activate the setting.
4. Click on the tick button on the right to save the setting.
5. Deploy the modified system configuration.



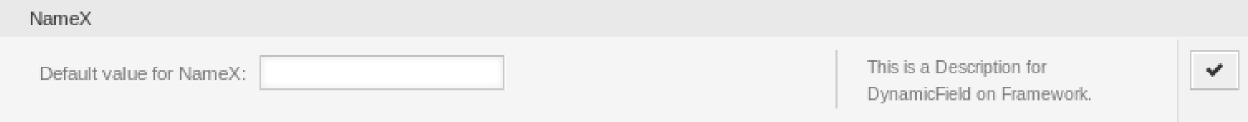
PreferencesGroups###DynamicField

Active	1
Block	Input
Data	[% Env("UserDynamicField_NameX")
Desc	iption for DynamicField on Framework.
Key	Default value for NameX
Label	NameX
Module	Kernel::Output::HTML::Preferences::G
PrefKey	UserDynamicField_NameX
PreferenceGroup	Miscellaneous
Prio	7000

Defines the config parameters of this item, to be shown in the preferences view. Please note: setting 'Active' to 0 will only prevent agents from editing settings of this group in their personal preferences, but will still allow administrators to edit the settings of another user's behalf. Use 'PreferenceGroup' to control in which area these settings should be shown in the user interface.

Fig. 49: Activate Dynamic Field in Personal Preferences

Click on your avatar on the top left corner, and select *Personal Preferences* → *Miscellaneous* to add a default value for the dynamic field.



NameX

Default value for NameX:

This is a Description for DynamicField on Framework.

Fig. 50: Dynamic Field in Personal Preferences

This setting is an example of how to create an entry in the user preferences screen to set an exclusive dynamic field `Name_X` default value for the selected user. The limitation of this setting is that it only permits the use of one dynamic field. If two or more fields will use this feature, it is necessary to create a custom XML configuration file to add more settings similar to this one.

Note: If more settings are added in a new XML each setting name needs to be unique in the system and different than `PreferencesGroups###DynamicField`. For example:

- `PreferencesGroups###101-DynamicField-Field1`
- `PreferencesGroups###102-DynamicField-Field2`
- `PreferencesGroups###My-Field1`
- `PreferencesGroups###My-Field2`

5.3 Generic Agent

Processing tickets require often a workflow. Let’s say “if-then” activities.

If specific conditions match like:

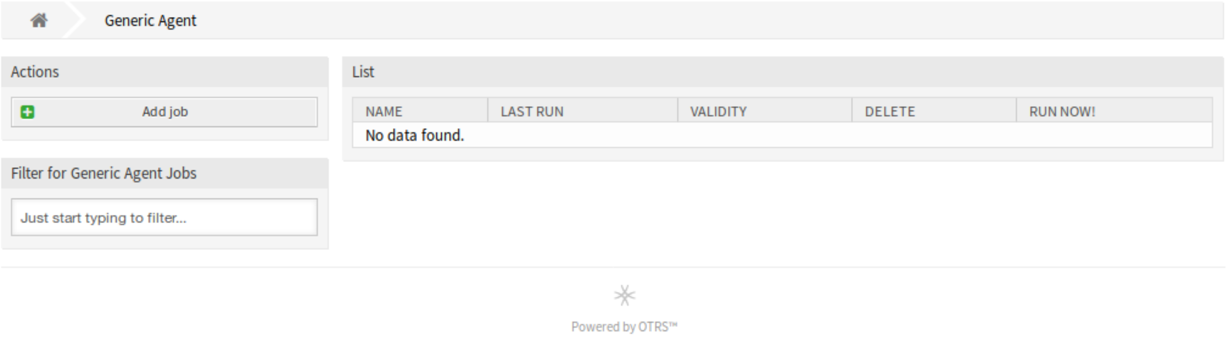
- A ticket is from one particular customer.
- A ticket is assigned to an appropriate queue.
- A ticket has a defined priority.
- A ticket contains defined keywords.

Outlined activities must be performed like changing the ticket priority, moving the ticket to another group, assigning a service to a ticket, and many more.

Also time-based activities can be required like cleaning up the spam-queue once a week.

OTRS supports this with the *Generic Agent*. Here, simple or compound time and event-based tasks are configurable in the OTRS front end without the requirement to learn a scripting language. Depending on search criteria, and time or event criteria, tickets will automatically be acted upon.

Use this screen to manage generic agent jobs in the system. A fresh OTRS installation contains no generic agent jobs by default. The generic agent job management screen is available in the *Generic Agent* module of the *Processes & Automation* group.



NAME	LAST RUN	VALIDITY	DELETE	RUN NOW!
No data found.				

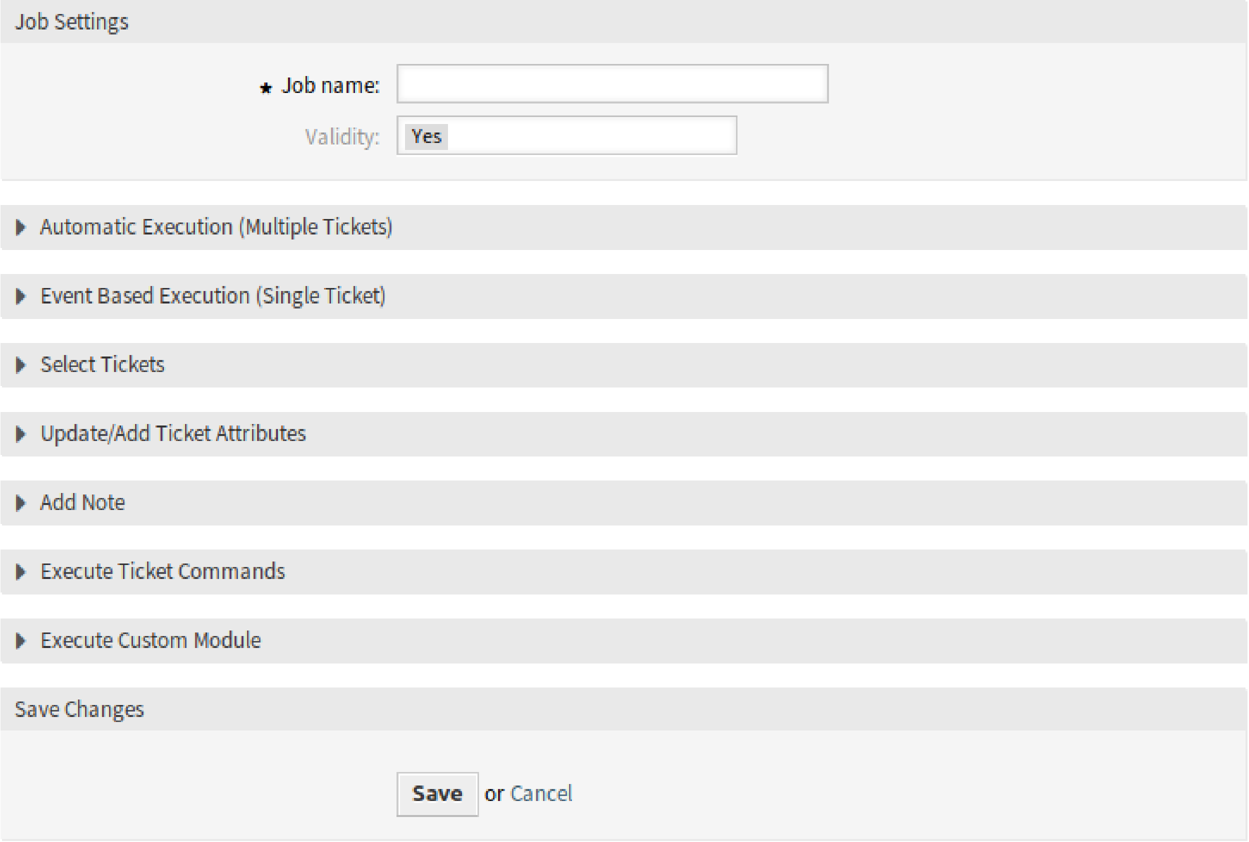
Fig. 51: Generic Agent Management Screen

5.3.1 Manage Generic Agent Jobs

To create a new generic agent job:

1. Click on the *Add Job* button in the left sidebar.
2. Fill in the required fields.

3. Click on the *Save* button.



Job Settings

★ Job name:

Validity: Yes

▸ Automatic Execution (Multiple Tickets)

▸ Event Based Execution (Single Ticket)

▸ Select Tickets

▸ Update/Add Ticket Attributes

▸ Add Note

▸ Execute Ticket Commands

▸ Execute Custom Module

Save Changes

Save or Cancel

Fig. 52: Create New Generic Agent Job Screen

To edit a generic agent job:

1. Click on a generic agent job in the list of generic agent jobs.
2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.

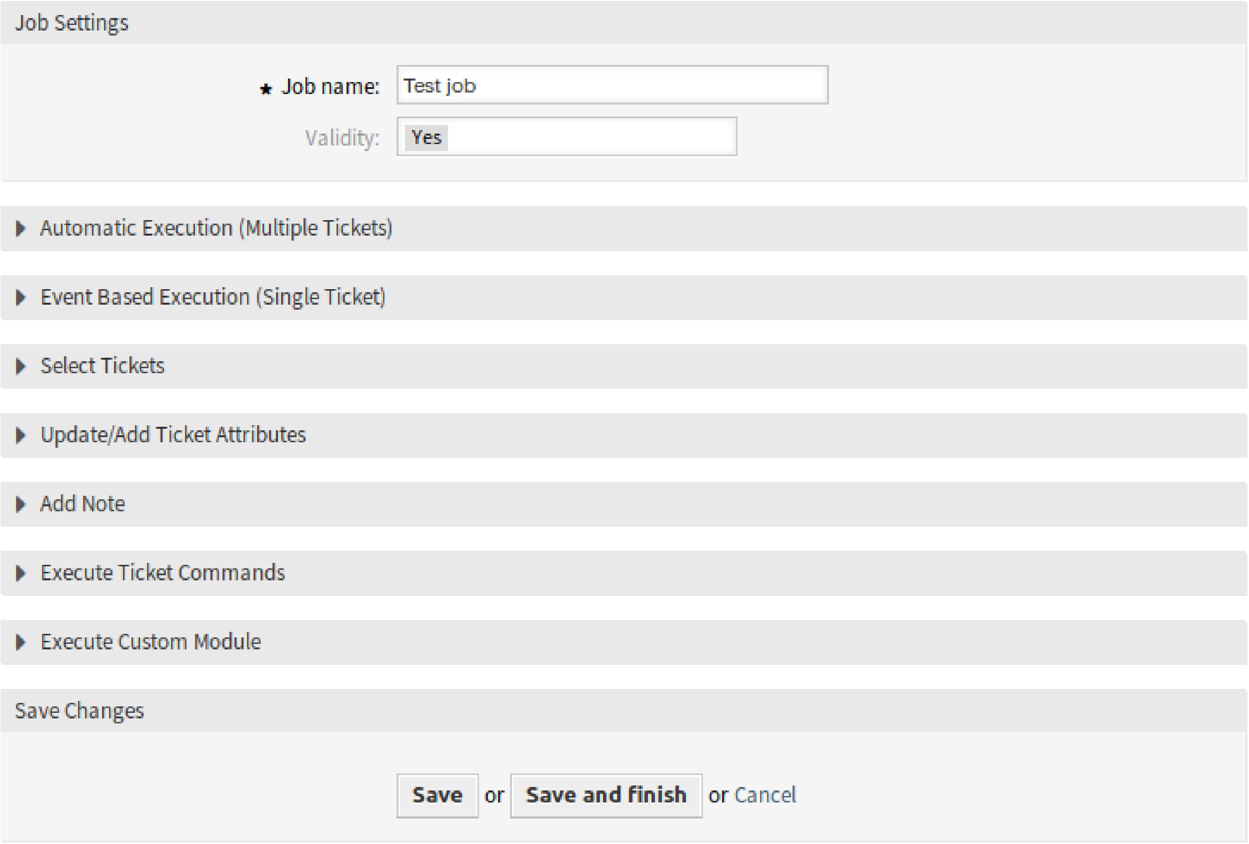
To delete a generic agent job:

1. Click on the trash icon in the fourth column of the overview table.
2. Click on the *Confirm* button.

Note: If several generic agent jobs are added to the system, use the filter box to find a particular generic agent job by just typing the name to filter.

5.3.2 Generic Agent Job Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.



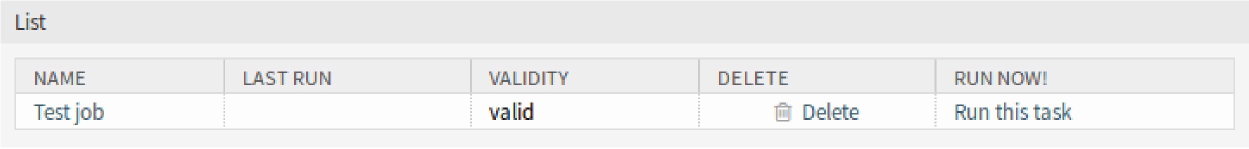
Job Settings

★ Job name: Test job

Validity: Yes

▸ Automatic Execution (Multiple Tickets)
▸ Event Based Execution (Single Ticket)
▸ Select Tickets
▸ Update/Add Ticket Attributes
▸ Add Note
▸ Execute Ticket Commands
▸ Execute Custom Module
Save Changes
Save
or
Save and finish
or Cancel

Fig. 53: Edit Generic Agent Job Screen

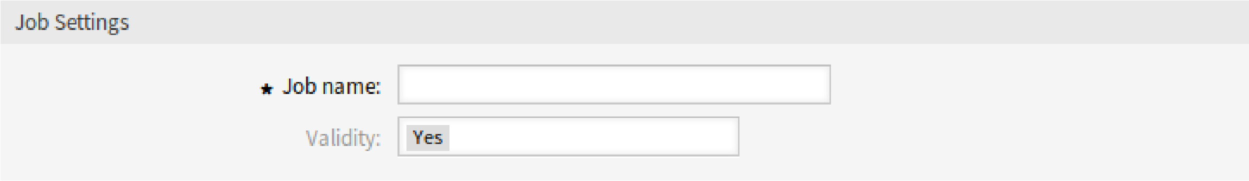


List

NAME	LAST RUN	VALIDITY	DELETE	RUN NOW!
Test job		valid	Delete	Run this task

Fig. 54: Delete Generic Agent Job Screen

General Job Settings



Job Settings

★ Job name:

Validity: Yes

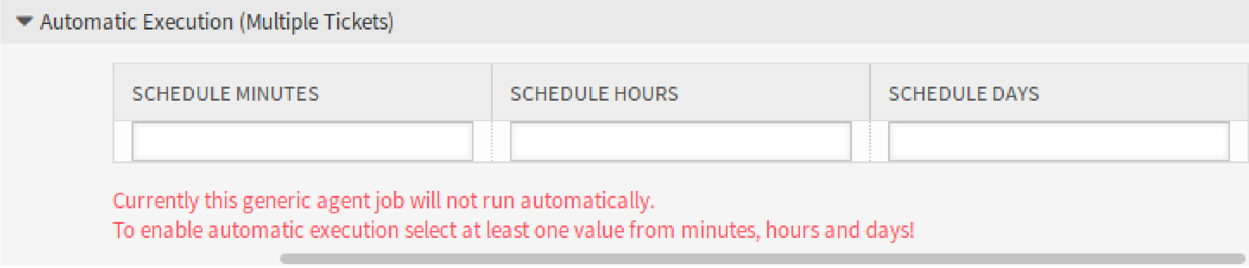
Fig. 55: Job Settings - General

Job Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Validity Set the validity of this resource. This resource can be used in OTRS only, if this field is set to *Yes*. Setting this field to *No* will disable the use of the resource.

Automatic Execution

Fill in this section to set the times for automatic execution of the job.



▼ Automatic Execution (Multiple Tickets)

SCHEDULE MINUTES	SCHEDULE HOURS	SCHEDULE DAYS

Currently this generic agent job will not run automatically.
To enable automatic execution select at least one value from minutes, hours and days!

Fig. 56: Job Settings - Automatic Execution

Schedule minutes Select the minutes in which the job has to be executed. For example if *10* is selected, the job will be executed in every hour in 00:10, 01:10, 02:10, etc.

Schedule hours Select the hours in which the job has to be executed. For example if *10* is selected for minutes and *02* is selected for hours, the job will be executed in every day in 02:10.

Schedule days Select the days in which the job has to be executed. For example if *10* is selected for minutes, *02* is selected for hours and *Fri* is selected for days, the job will be executed in every week in Friday 02:10.

Event Based Execution

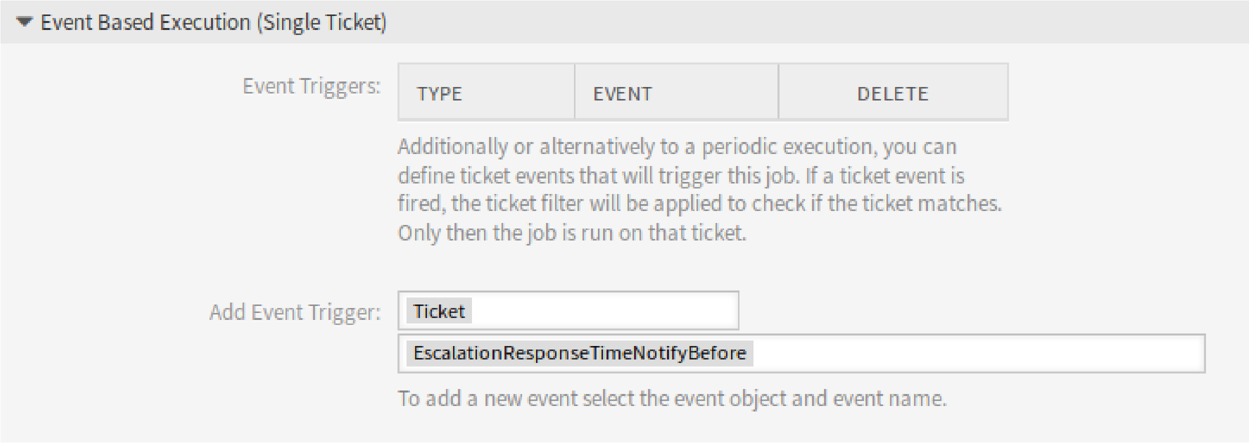
Fill in this section to set events that trigger the execution of the job.

Event Triggers This is a list of already added events. If no events are added yet, the list contains no elements. Elements can be deleted by clicking on the trash icon in the last column.

Add Event Trigger Select an object and a related event that will trigger the job.

Select Tickets

This section contains all the attributes, that you can use to select the affected tickets.



▼ Event Based Execution (Single Ticket)

Event Triggers:

TYPE	EVENT	DELETE

Additionally or alternatively to a periodic execution, you can define ticket events that will trigger this job. If a ticket event is fired, the ticket filter will be applied to check if the ticket matches. Only then the job is run on that ticket.

Add Event Trigger: Ticket

EscalationResponseTimeNotifyBefore

To add a new event select the event object and event name.

Fig. 57: Job Settings - Event Based Execution

Fields are self-explained, so there is no additional explanation here.

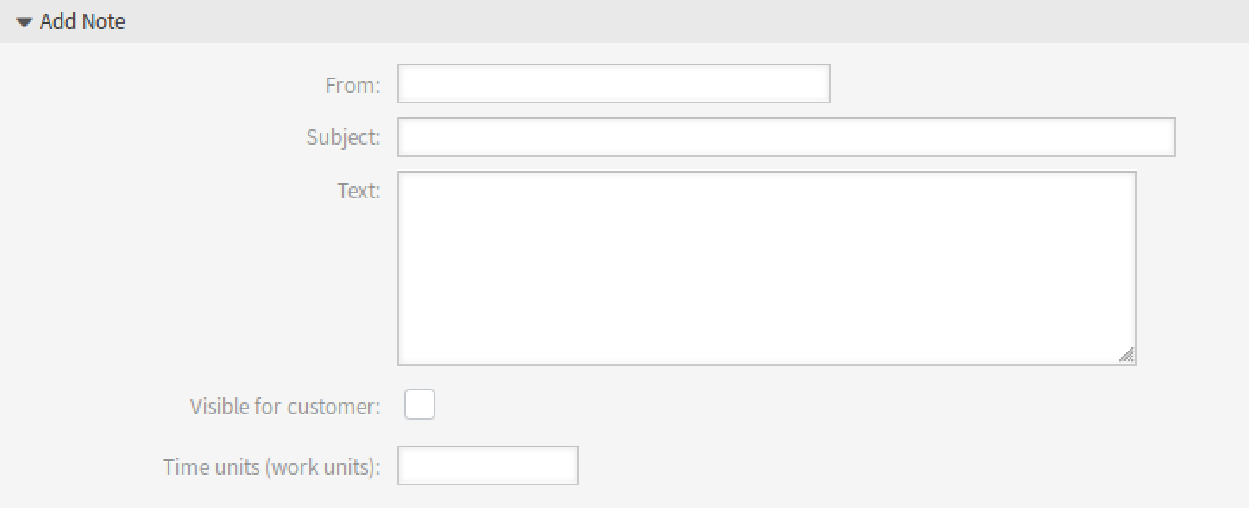
Update/Add Ticket Attributes

Fill in this section to update and/or add ticket attributes.

Fields are self-explained, so there is no additional explanation here.

Add Note

Fill in this section to add a note to tickets that are affected by job execution.



▼ Add Note

From:

Subject:

Text:

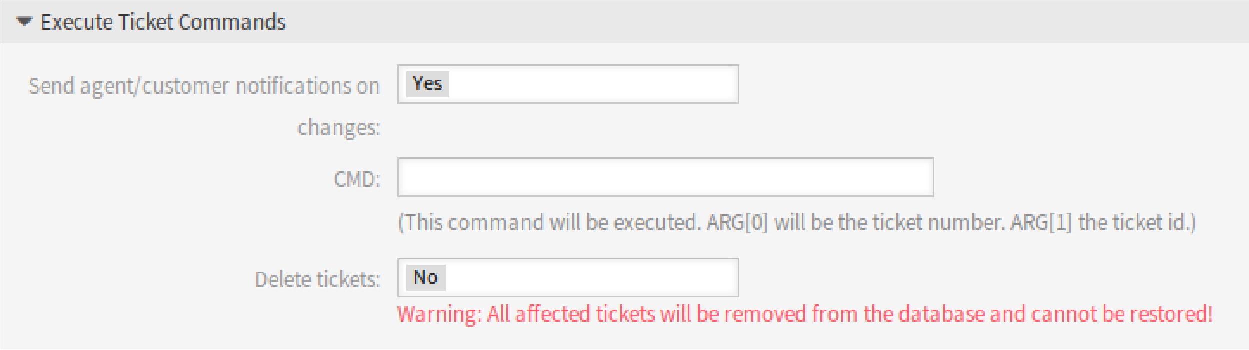
Visible for customer: □

Time units (work units):

Fig. 58: Job Settings - Add Note

Execute Ticket Commands

Fill in this section, if you would like to make execute a custom command with the job.



▼ Execute Ticket Commands

Send agent/customer notifications on changes: Yes

CMD:

(This command will be executed. ARG[0] will be the ticket number. ARG[1] the ticket id.)

Delete tickets: No

Warning: All affected tickets will be removed from the database and cannot be restored!

Fig. 59: Job Settings - Execute Ticket Commands

Send agent/customer notifications on changes If *Yes* is selected, a notification will be sent to agents and customers about ticket changes.

CMD Enter a command here, that will be executed. ARG[0] will be the ticket number. ARG[1] the ticket ID.

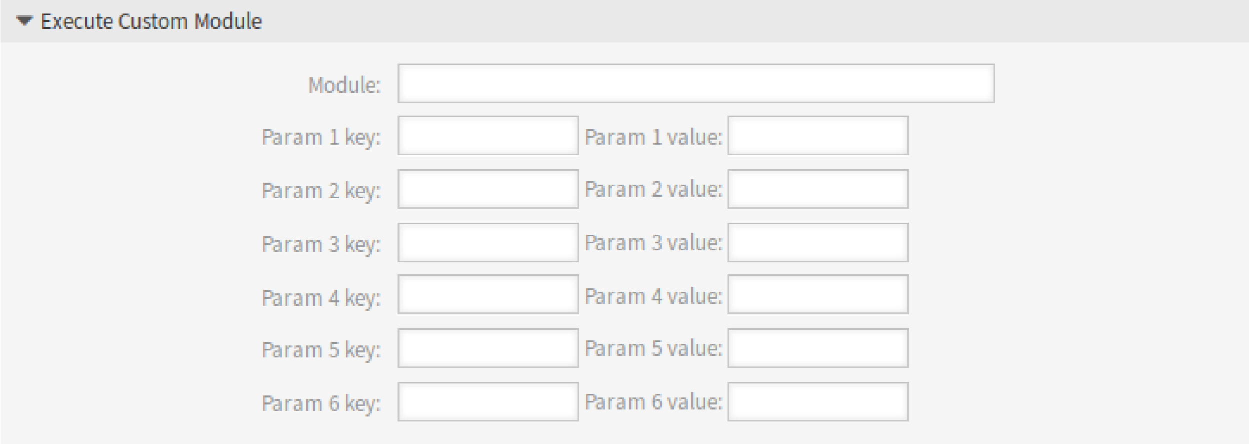
> **Note:** Use `::` as directory separator, if the value contains a path.

Delete tickets If *Yes* is selected, the generic agent job will delete the matched tickets.

> **Warning:** All affected tickets will be removed from the database and cannot be restored!

Execute Custom Module

Fill in this section, if you would like to make execute a custom module with the job.



▼ Execute Custom Module

Module:

Param 1 key: Param 1 value:

Param 2 key: Param 2 value:

Param 3 key: Param 3 value:

Param 4 key: Param 4 value:

Param 5 key: Param 5 value:

Param 6 key: Param 6 value:

Fig. 60: Job Settings - Execute Custom Module

Module This is the path for the module to be executed.

> **Note:** Use `::` as directory separator, if the value contains a path.

Param key Enter the key of the parameter, that should be passed to the module.

Param value Enter the value of the parameter, that should be passed to the module.

5.4 Process Management

Maximizing performance while minimizing human error is a requirement of organizations of all sizes. This need is covered by defined processes and workflows, for recurring tasks. Ensuring all required information is available in the right place, and contacts are informed about their responsibilities like adding information, approving requests, etc.

OTRS supports this requirement by process management. Process tickets help by using the required mandatory and optional fields (see *Dynamic Fields*) that information is not forgotten upon ticket creation or during later steps of the process. Process tickets are simple to handle for customer users and agents, so no intensive training is required.

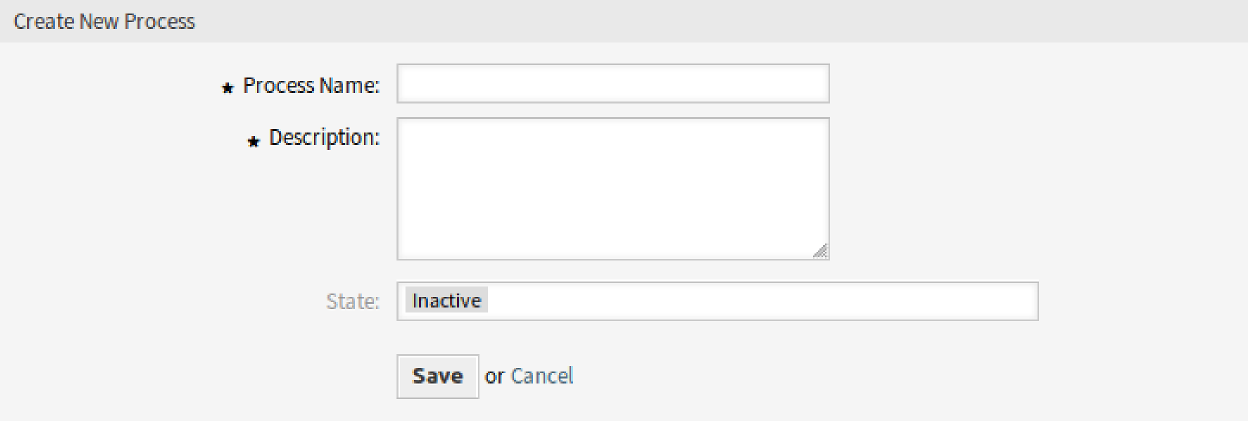
Processes are designed completely and efficiently within the OTRS front end to fit the requirements of your organization.

Use this screen to manage processes in the system. The process management screen is available in the *Process Management* module of the *Processes & Automation* group.

5.4.1 Manage Processes

To create a new process:

1. Click on the *Create New Process* button in the left sidebar.
2. Fill in the required fields.
3. Click on the *Save* button.
4. Add activities, user task activity dialogs, sequence flows and sequence flow actions.
5. Set *State* to *Active*.
6. Deploy all processes.


Create New Process

* Process Name:

* Description:

State: Inactive

Fig. 61: Create New Process Screen

To edit a process:

1. Click on a process in the list of processes.

2. Modify the fields and the process path.
3. Click on the *Save* or *Save and finish* button.
4. Deploy all processes.

To copy a process:

1. Click on the copy icon in the fifth column of list of processes.
2. Click on the newly created process to edit it.

To delete a process:

1. Click on a process in the list of processes.
2. Set the *State* option to *Inactive*.
3. Click on the *Save* button. A new *Delete Inactive Process* button will appear in the left sidebar.
4. Click on the *Delete Inactive Process* button.
5. Click on the *Delete* button in the confirmation screen.
6. Deploy all processes.

Warning: Processes are written into file in Perl format. Without deploying, all processes are still in this cache file even if they are deleted or the *State* option is set to *Inactive* or *FadeAway*. Don't forget to deploy all processes after modifications!

Give some time for agents to complete the running process tickets before the process will be deleted. It is possible to mark a process for deletion, i. e. set the process as not to be selected anymore. Process states can be:

Active Processes can be used in new process tickets.

FadeAway Processes cannot be selected any more for new tickets, but existing tickets still can use the process.

Inactive Processes are deactivated and cannot be used for new or existing tickets.

To deploy all processes:

1. Click on the *Deploy All Processes* button in the left sidebar.

Note: New or modified processes have to deploy in order to make affect the behavior of the system. Setting the *State* option to *Active* just indicates, which processes should be deployed.

To export a process:

1. Click on the export icon in the fourth column of list of processes.
2. Choose a location in your computer to save the `Export_ProcessEntityID_xxx.yml` file.

Warning: The exported file contains only the process itself, and doesn't contain the *Queues*, *Agents*, *Dynamic Fields*, etc. needed for the process.

To import a process:

1. Click on the *Browse…* button of the *Configuration Import* widget in the left sidebar.

2. Select a previously exported `.yml` file.
3. Click on the *Import process configuration* button.
4. Deploy all processes.

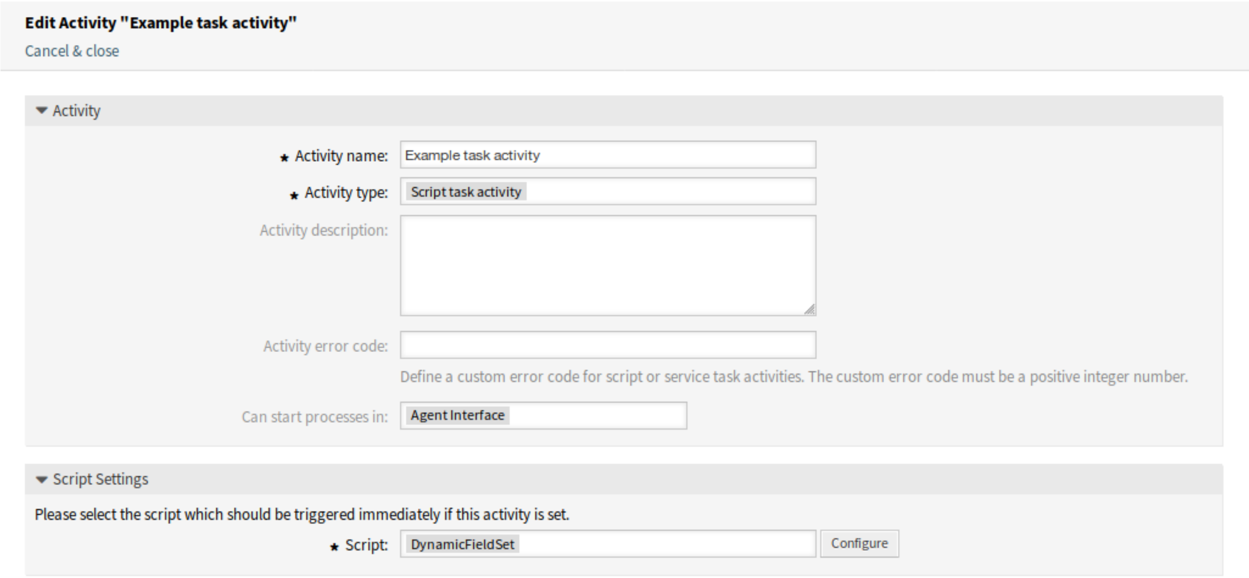
Note: Before import a process, it is still necessary to create all *Queues*, *Agents* and *Dynamic Fields*, as well as to set *System Configuration*, that are needed by each process before the import. If the process requires the use of *Access Control Lists (ACL)* those are also needed to be set manually.

Note: If several processes are added to the system, use the filter box to find a particular process by just typing the name to filter.

5.4.2 Process Elements

The element names of the process modeler have been adapted to the *Business Process Model and Notation* (BPMN) ISO naming convention. The following task activities can be used as basic elements of the process.

Script task activity Script task activity is executed by the process management module and it can set dynamic field values or manage tickets automatically.



Edit Activity "Example task activity"

★ Activity name: Example task activity

★ Activity type: Script task activity

Activity description:

Activity error code:

Define a custom error code for script or service task activities. The custom error code must be a positive integer number.

Can start processes in: Agent Interface

Please select the script which should be triggered immediately if this activity is set.

★ Script: DynamicFieldSet

Fig. 62: Script Task Activity Window

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Activity name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces.

Activity type * The following types of task activities can be used:

- Script task activity (selected for now)
- Service task activity

- User task activity

Activity description Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity.

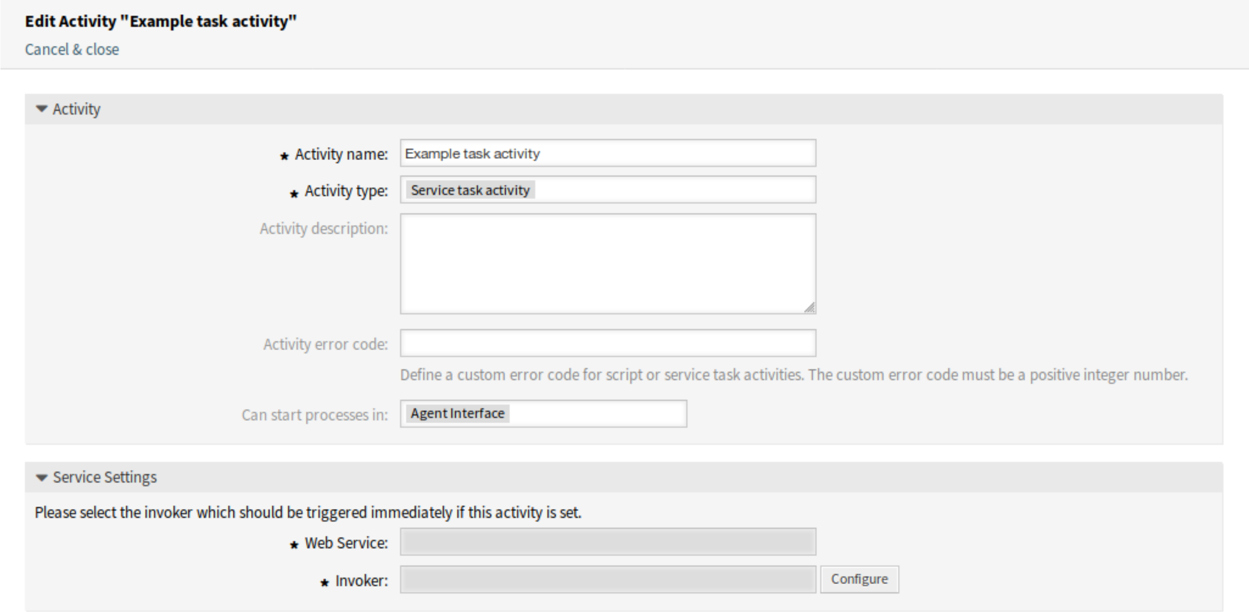
Activity error code Define a custom error code for script or service task activities. The custom error code must be a positive integer number.

Can start processes in Defines, where can the processes be started by the agents or customer users. A process can be started in the following interfaces:

- Agent Interface
- Agent and External Interface
- External Interface

Script * In this drop-down can be selected which script should be triggered immediately if the activity is set. Click on the *Configure* button to add parameters (key-value pairs) for the script.

Service task activity Service task activity uses a web service to complete the task.



Edit Activity "Example task activity"

Cancel & close

Activity

★ Activity name: Example task activity

★ Activity type: Service task activity

Activity description:

Activity error code:

Define a custom error code for script or service task activities. The custom error code must be a positive integer number.

Can start processes in: Agent Interface

Service Settings

Please select the invoker which should be triggered immediately if this activity is set.

★ Web Service:

★ Invoker:

Fig. 63: Service Task Activity Window

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Activity name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces.

Activity type * The following types of task activities can be used:

- Script task activity
- Service task activity (selected for now)
- User task activity

Activity description Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity.

Activity error code Define a custom error code for script or service task activities. The custom error code must be a positive integer number.

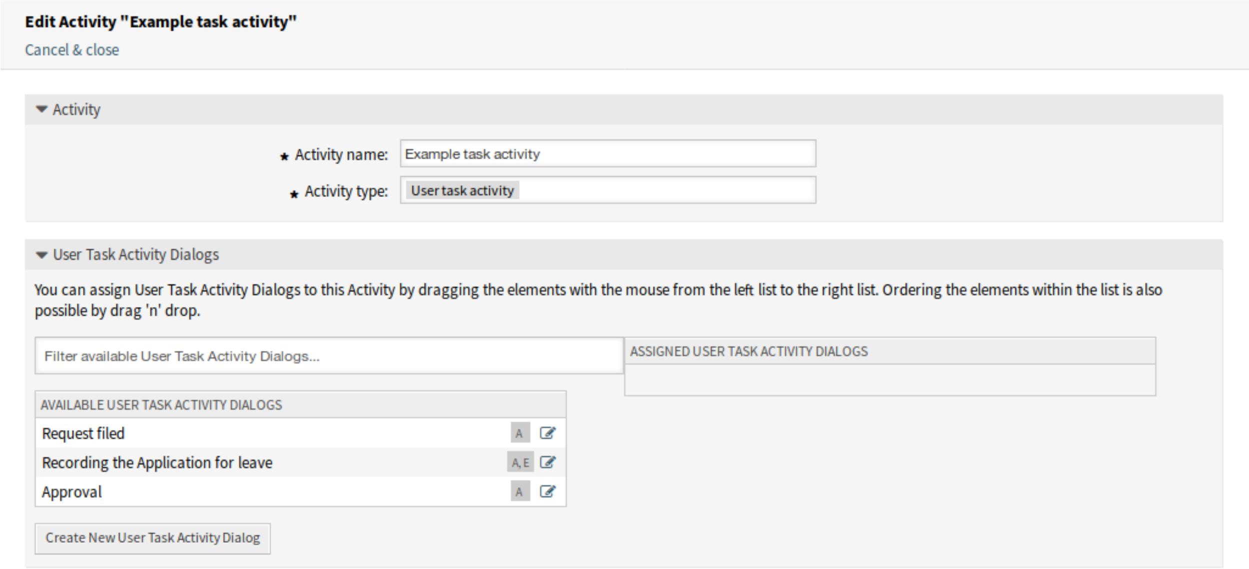
Can start processes in Defines, where can the processes be started by the agents or customer users. A process can be started in the following interfaces:

- Agent Interface
- Agent and External Interface
- External Interface

Web Service * Select one of the *Web Services* from the drop-down list.

Invoker * Select an invoker for the web service. Click on the *Configure* button to add parameters for the invoker.

User task activity User task activity can be used when the task is being done by an agent or a customer user.



Edit Activity "Example task activity"

Cancel & close

Activity

* Activity name: Example task activity

* Activity type: User task activity

User Task Activity Dialogs

You can assign User Task Activity Dialogs to this Activity by dragging the elements with the mouse from the left list to the right list. Ordering the elements within the list is also possible by drag 'n' drop.

Filter available User Task Activity Dialogs...

ASSIGNED USER TASK ACTIVITY DIALOGS

AVAILABLE USER TASK ACTIVITY DIALOGS	
Request filed	A
Recording the Application for leave	A, E
Approval	A

Create New User Task Activity Dialog

Fig. 64: User Task Activity Window

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Activity name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces.

Activity type * The following types of task activities can be used:

- Script task activity
- Service task activity
- User task activity (selected for now)

User Task Activity Dialogs You can assign user task activity dialogs to this activity by dragging the elements with the mouse from the left list to the right list. Ordering the elements within the list is also possible by drag and drop.

Click on the *Create New User Task Activity Dialog* button to create new dialogs.

5.4.3 Example process

Processes are more complex than other resources in OTRS. To create a process, you need to do several steps. The following chapters shows you, how to define a process from the specification and create the needed resources. Let’s see an example to make it more demonstrative. We will define a book order process.

Process Specification

The book order process has four states.

Recording the demand Before an order will be placed, the demand for literature by an employee will be recorded. The following book is needed in our example:

```
Title: Prozessmanagement für Dummies
Autor: Thilo Knuppertz
ISBN: 3527703713
```

Approval by manager The head of the employee’s department needs to decide on the order. In case of a denial, a reason should be recorded by the manager. In case of approval, the order is passed to the purchasing department.

Processing by purchasing department Purchasing now has the task to find out where the book can be ordered with the best conditions. If it is out of stock, this can be recorded in the order. In case of a successful order purchasing will record the supplier, the price and the delivery date.

Processing by the mail room The shipment will arrive at the company. The incoming goods department checks the shipment and records the date of receipt. Now the employee will be informed that their order has arrived and is ready to be collected.

Introduce The Process Elements

If we assume that a ticket acts in this workflow like an accompanying document that can receive change notes, we already have a clear picture of process tickets.

From the analysis of the example process we can identify the following necessary items:

- Possibility to record data, let’s call this *user task activity dialog*.
- Check which can react to changed data automatically, let’s call this *sequence flow*.
- Change which can be applied to a process ticket after successful transitions of a process ticket, let’s call this *sequence flow action*.
- A possibility to offer more than just one user task activity dialog to be available. In our example this is needed when the manager must have the choice between *Approve* and *Deny*. Let’s call this *activity*.

Now, with activities, user task activity dialogs, sequence flows and sequence flow actions we have the necessary tools to model the individual steps of our example. What is still missing is an area where for each workflow the order of the steps can be specified. Let’s call this *process*.

Create Necessary Resources

Before the creation of the process and its parts is necessary to prepare the system. We need to define some *Queues*, *Agents* and *Dynamic Fields* as well as set some *System Configuration* settings.

Create the following *Queues*:

- Management
- Employees
- Purchasing
- Post office

Create the following *Agents*:

- Manager
- Employee

Create the following *Dynamic Fields*:

Object	Type	Name	Label	Possible values
Ticket	Text	`Title`	Title	
Ticket	Text	`Author`	Author	
Ticket	Text	`ISBN`	ISBN	
Ticket	Dropdown	`Status`	Status	• Approval • Approval denied • Approved • Order denied • Order placed • Shipment received
Ticket	Text	`Supplier`	Supplier	
Ticket	Text	`Price`	Price	
Ticket	Date	`DeliveryDate`	Delivery date	
Ticket	Date	`DateOfReceipt`	Date of receipt	

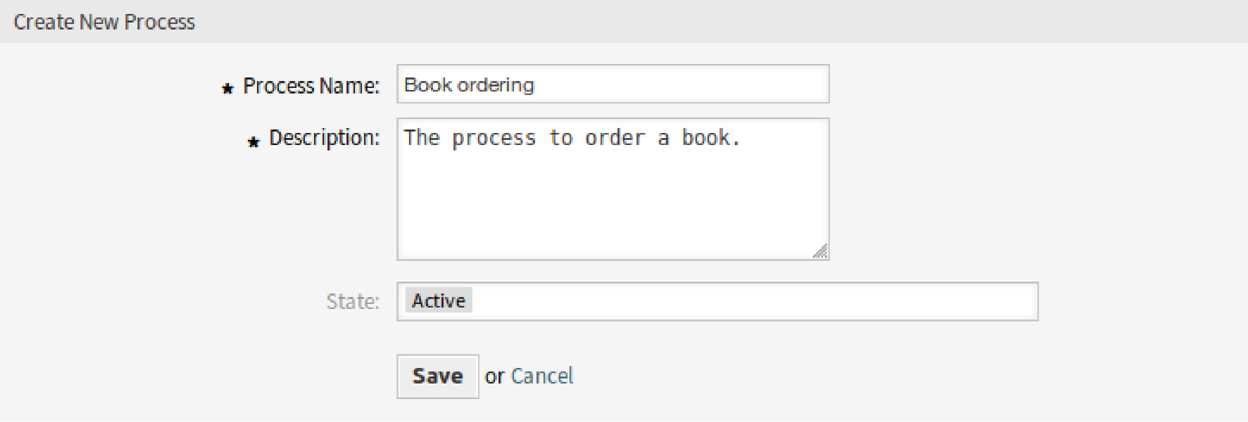
Set the following *System Configuration* settings:

- Ticket::Responsible
  - Enabled
- AgentFrontend::TicketDetailView::Widget::BusinessProcessInformation###DynamicField
  - `Author` → *1 - Enabled*
  - `DateOfReceipt` → *1 - Enabled*
  - `DeliveryDate` → *1 - Enabled*
  - `ISBN` → *1 - Enabled*
  - `Price` → *1 - Enabled*
  - `Status` → *1 - Enabled*
  - `Supplier` → *1 - Enabled*
  - `Title` → *1 - Enabled*
- AgentFrontend::TicketDetailView::Widget::BusinessProcessInformation###DynamicFieldGroups
  - `Book` → `Title,Author,ISBN`

   - `General` → `Status`
   - `Order` → `Price`, `Supplier`, `DeliveryDate`
   - `Shipment` → `DateOfReceipt`

Note: Don' t forget to deploy the modified system configuration settings.

Now, go back to the *Process Management* screen and click on the *Create New Process*. Fill in the required fields.



Create New Process

★ Process Name: Book ordering

★ Description: The process to order a book.

State: Active

Fig. 65: Book Ordering - Create New Process

The new process is created. You can add some process element now.

Create User Task Activity Dialogs

Click on the *User Task Activity Dialogs* item in the *Available Process Elements* widget in the left sidebar. This action will expand the *User Task Activity Dialogs* options and will collapse all others doing an accordion like effect. Click on the *Create New User Task Activity Dialog* button.

In the opened popup screen fill in the *Dialog Name* as well as the *Description (short)* fields. For this example we will leave all other fields as the default.

To assign fields to the user task activity dialog simple drag the required field from the *Available Fields* pool and drop into the *Assigned Fields* pool. The order in the *Assigned Fields* pool is the order as the fields will have in the screen. To modify the order simply drag and drop the field within the pool to rearrange it in the correct place.

In this example we will use:

- `Article` field for comments.
- `DynamicField_Title`, `DynamicField_Author`, `DynamicField_ISBN` fields for the data to be collected for the order.
- `DynamicField_Status` with the possibility to choose *Approval*.

Drag these fields from the *Available Fields* pool and drop into the *Assigned Fields* pool.

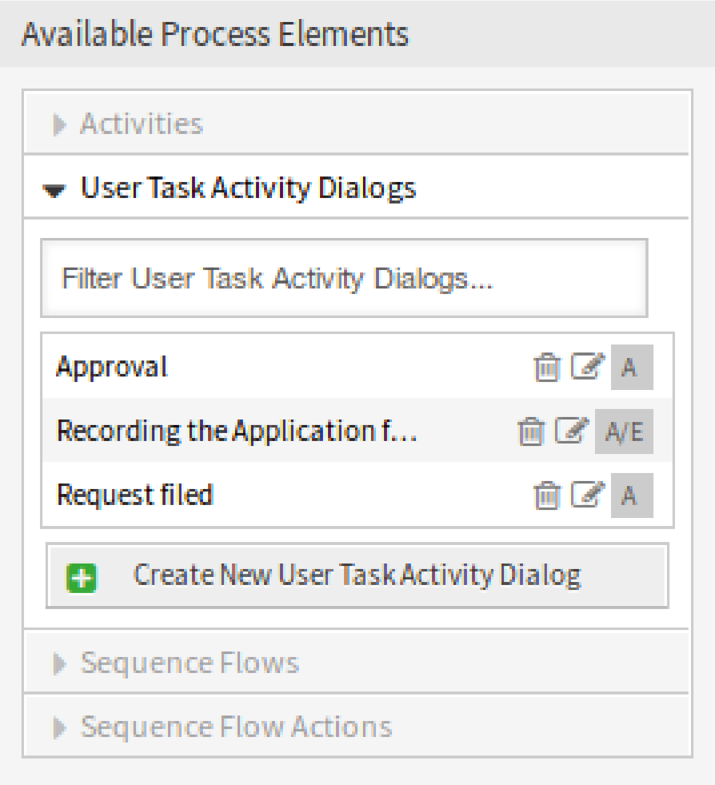
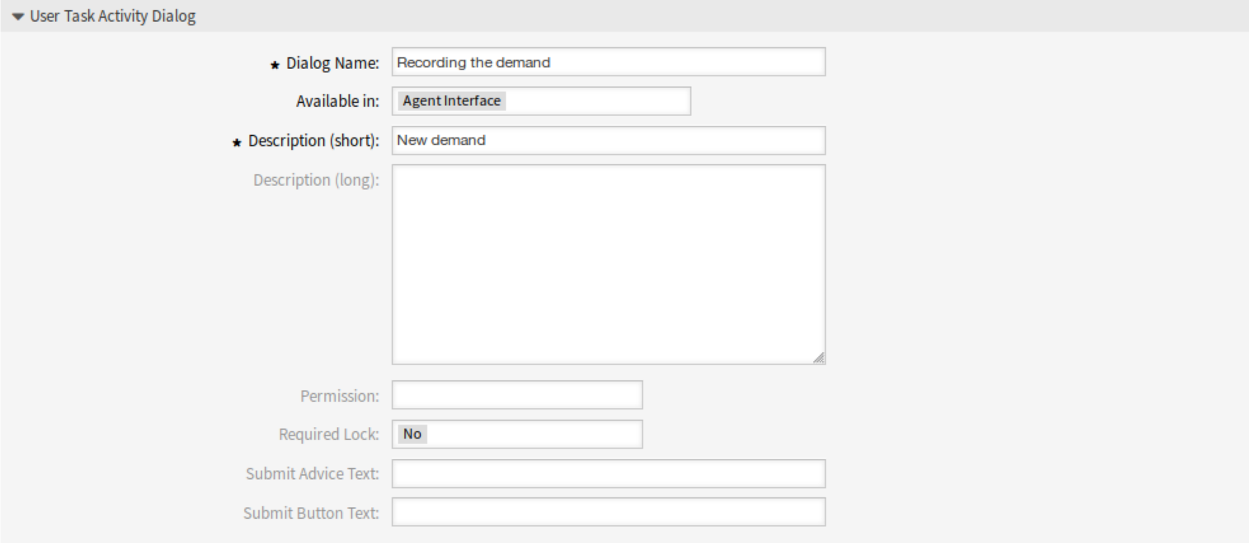

Available Process Elements
Activities
User Task Activity Dialogs
Filter User Task Activity Dialogs...
Approval
A
Recording the Application f...
A/E
Request filed
A
Create New User Task Activity Dialog
Sequence Flows
Sequence Flow Actions

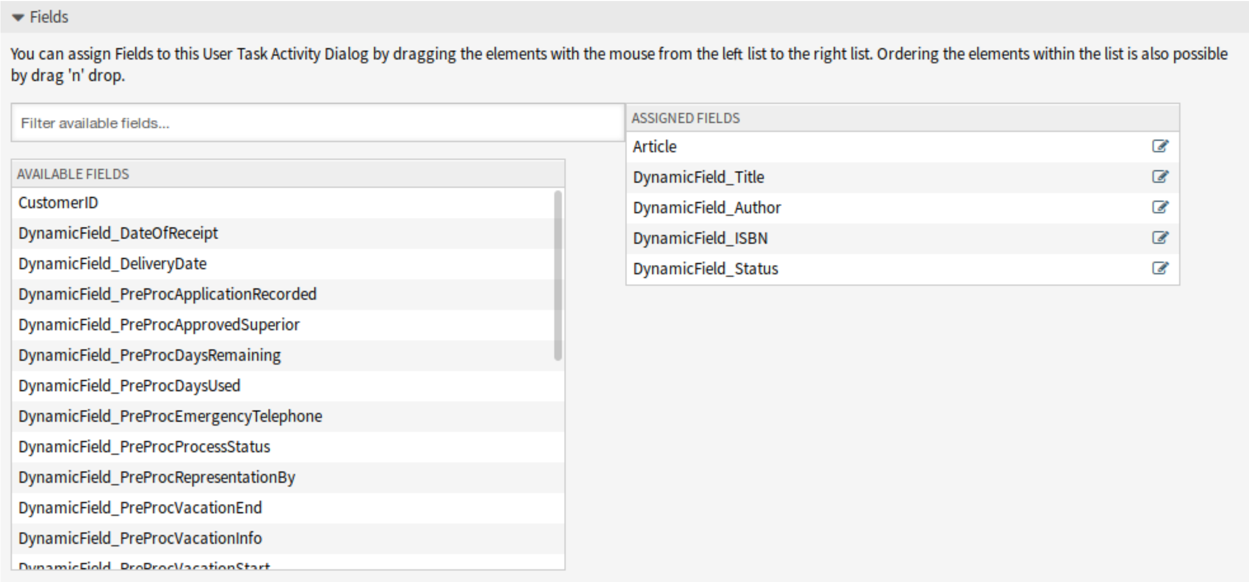
Fig. 66: Book Ordering - User Task Activity Dialogs



* Dialog Name:	Recording the demand
Available in:	Agent Interface
* Description (short):	New demand
Description (long):	
Permission:	
Required Lock:	No
Submit Advice Text:	
Submit Button Text:	

Fig. 67: Book Ordering - Add User Task Activity Dialog

Note: In this screen all dynamic fields has the prefix `DynamicField_` as in `DynamicField_Title`. Do not confuse with the field `Title` that is the ticket title.



You can assign Fields to this User Task Activity Dialog by dragging the elements with the mouse from the left list to the right list. Ordering the elements within the list is also possible by drag 'n' drop.

AVAILABLE FIELDS

- CustomerID
- DynamicField_DateOfReceipt
- DynamicField_DeliveryDate
- DynamicField_PreProcApplicationRecorded
- DynamicField_PreProcApprovedSuperior
- DynamicField_PreProcDaysRemaining
- DynamicField_PreProcDaysUsed
- DynamicField_PreProcEmergencyTelephone
- DynamicField_PreProcProcessStatus
- DynamicField_PreProcRepresentationBy
- DynamicField_PreProcVacationEnd
- DynamicField_PreProcVacationInfo
- DynamicField_PreProcVacationStart

ASSIGNED FIELDS

- Article
- DynamicField_Title
- DynamicField_Author
- DynamicField_ISBN
- DynamicField_Status

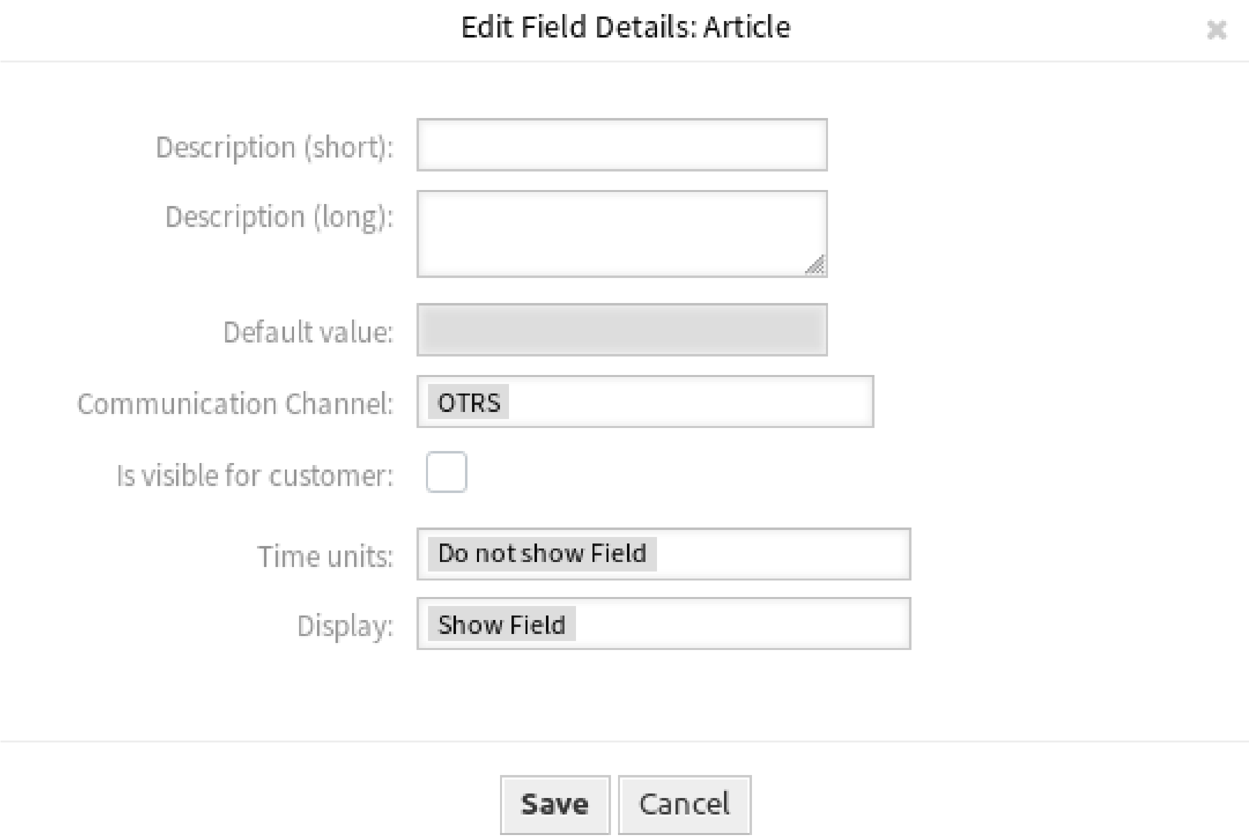
Fig. 68: Book Ordering - Add User Task Activity Dialog Fields

As soon as the fields are dropped into the *Assigned Fields* pool another popup screen is shown with some details about the field. We will leave the default options and only for `Article` fields we should make sure that the *Communication Channel* field is set to *OTRS* and that the *Is visible for customer* is not checked.

After all fields are filled in, click on the *Save and finish* button to save the changes and go back to the project management screen.

Create the following user task activity dialogs with fields:

- *Recording the demand* (already created before)
  - `Article` field for comments.
  - `DynamicField_Title`, `DynamicField_Author`, `DynamicField_ISBN` fields for the data to be collected for the order.
  - `DynamicField_Status` with the possibility to choose *Approval*.
- *Approval denied*
  - `Article` field for comments.
  - `DynamicField_Status` with the possibility to choose *Approval denied*.
- *Approved*
  - `DynamicField_Status` with the possibility to choose *Approved*.
- *Order denied*
  - `Article` field for comments.
  - `DynamicField_Status` with the possibility to choose *Order denied*.



Edit Field Details: Article

Description (short):

Description (long):

Default value:

Communication Channel: OTRS

Is visible for customer: □

Time units: Do not show Field

Display: Show Field

Fig. 69: Book Ordering - Edit User Task Activity Dialog Fields

- *Order placed*
  - `DynamicField_Supplier`, `DynamicField_Price`, `DynamicField_DeliveryDate` fields for purchasing.
  - `DynamicField_Status` with the possibility to choose *Order placed*.
- *Shipment received*
  - `DynamicField_DateOfReceipt` for the mail room.
  - `DynamicField_Status` with the possibility to choose *Shipment received*.

Create Sequence Flows

Click on the *Sequence Flows* item in the *Available Process Elements* widget in the left sidebar. This action will expand the *Sequence Flows* options and will collapse all others doing an accordion like effect. Click on the *Create New Sequence Flow* button.

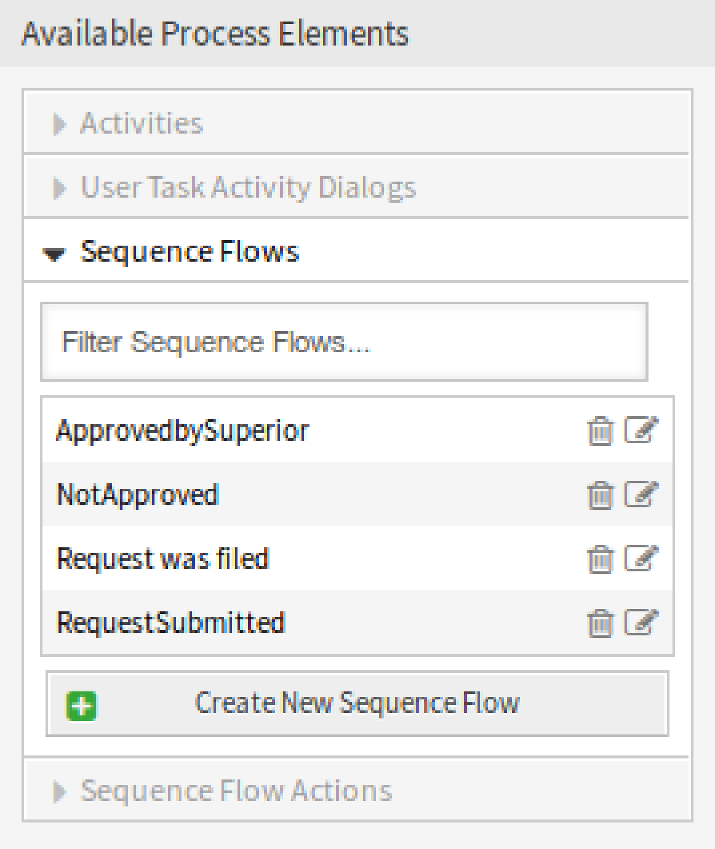
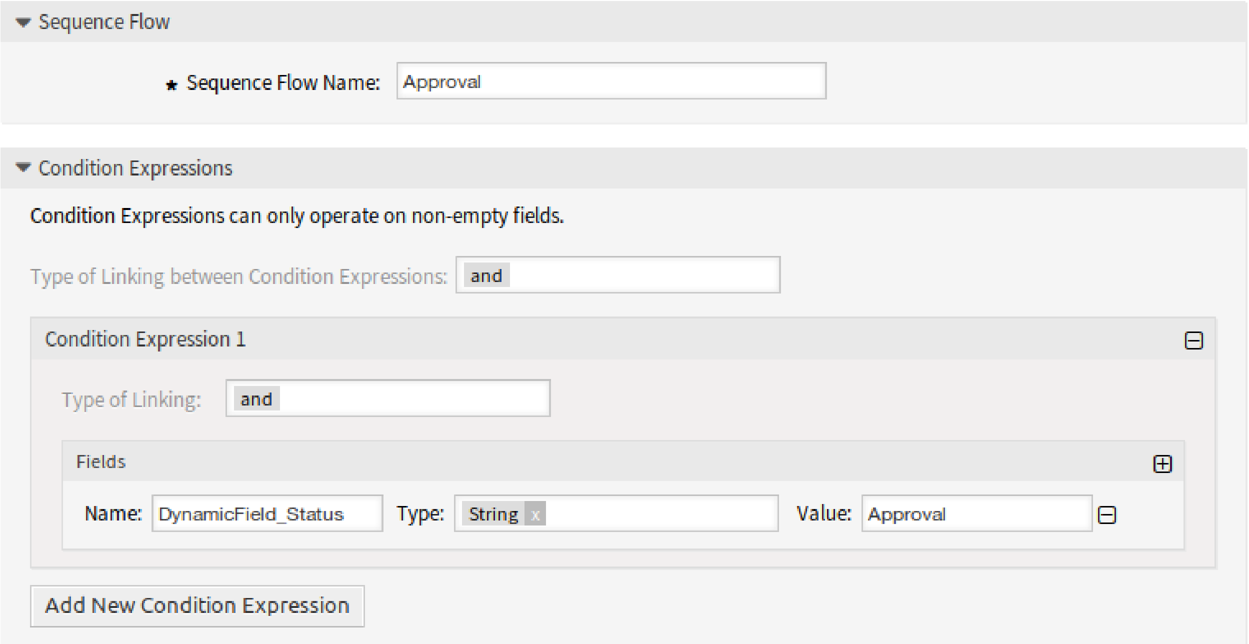


Fig. 70: Book Ordering - Sequence Flows

In the opened popup screen fill in the *Sequence Flow Name*. For this example in the *Condition Expressions* we will use just one condition expression and just one field. For both we can leave the *Type of Linking* as *and* and we will use the filed match type value as *String*.

After all fields are filled in, click on the *Save and finish* button to save the changes and go back to the project management screen.

Create the following sequence flows:



Sequence Flow

★ Sequence Flow Name: Approval

Condition Expressions

Condition Expressions can only operate on non-empty fields.

Type of Linking between Condition Expressions: and

Condition Expression 1

Type of Linking: and

Fields

Name: DynamicField_Status Type: String Value: Approval

Fig. 71: Book Ordering - Add Sequence Flow

- *Approval* (already created before)

  Check if the `DynamicField_Status` is set to *Approval*.

- *Approval denied*

  Check if the `DynamicField_Status` field is set to *Approval denied*.

- *Approved*

  Check if the `DynamicField_Status` field is set to *Approved*.

- *Order denied*

  Check if the `DynamicField_Status` field is set to *Order denied*.

- *Order placed*

  Check if the `DynamicField_Status` field is set to *Order placed*.

- *Shipment received*

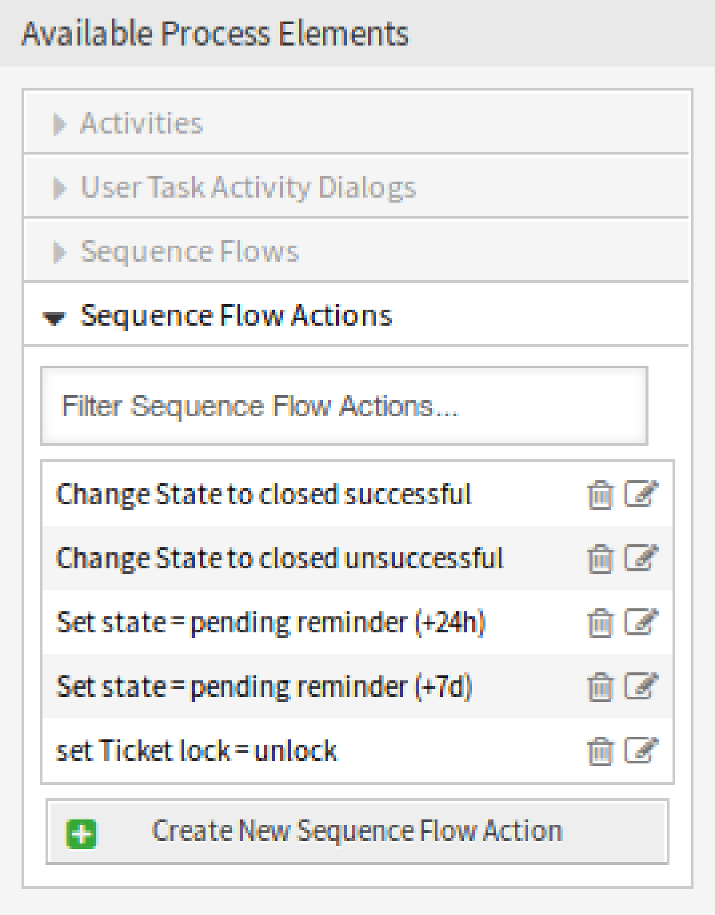
  Check if the `DynamicField_Status` field is set to *Shipment received*.

Create Sequence Flow Actions

Click on the *Sequence Flow Actions* item in the *Available Process Elements* widget in the left sidebar. This action will expand the *Sequence Flow Actions* options and will collapse all others doing an accordion like effect. Click on the *Create New Sequence Flow Action* button.

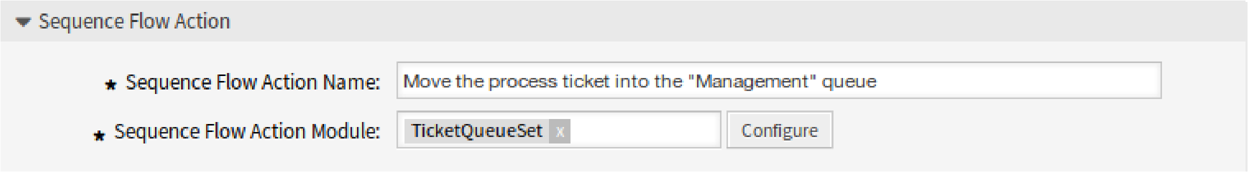
In the opened popup screen fill in the *Sequence Flow Action Name* and the *Sequence Flow Action module* then click on the *Save* button. A new *Configure* button will appear next to the module field.

Click on the *Configure* button and add the needed configuration parameter keys and values.


Available Process Elements
Activities
User Task Activity Dialogs
Sequence Flows
Sequence Flow Actions
Filter Sequence Flow Actions...
Create New Sequence Flow Action

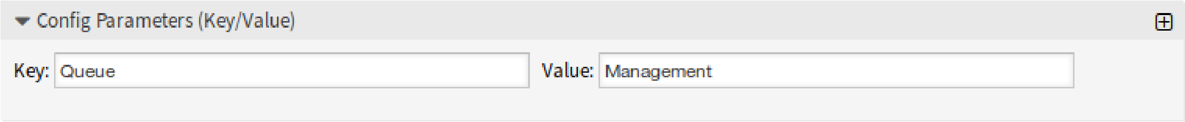
- Change State to closed successful
- Change State to closed unsuccessful
- Set state = pending reminder (+24h)
- Set state = pending reminder (+7d)
- set Ticket lock = unlock

Fig. 72: Book Ordering - Sequence Flow Actions



* Sequence Flow Action Name:	Move the process ticket into the "Management" queue
* Sequence Flow Action Module:	TicketQueueSet

Fig. 73: Book Ordering - Add Sequence Flow Action



Key: Queue Value: Management

Fig. 74: Book Ordering - Sequence Flow Action Parameters

After all fields are filled in, click on the *Save and finish* button to save the changes and go back to the project management screen.

See also:

Each module has its own and different parameters. Please review the module documentation to learn all require and optional parameters.

- DynamicFieldSet
- TicketArticleCreate
- TicketCreate
- TicketCustomerSet
- TicketLockSet
- TicketOwnerSet
- TicketQueueSet
- TicketResponsibleSet
- TicketSLASet
- TicketSendEmail
- TicketServiceSet
- TicketStateSet
- TicketTitleSet
- TicketTypeSet

All sequence flow action modules are located in the legacy named `Kernel/System/ProcessManagement/TransitionAction`.

Create the following sequence flow actions:

- *Move the process ticket into the "Management" queue* (already created before)

  To be executed when the sequence flow *Approval* applied.

- *Change ticket responsible to "Manager"*

  To be executed when the sequence flow *Approval* applied.

- *Move process ticket into the "Employees" queue*

  To be executed when:

  - The sequence flow *Approval denied* applied.
  - The sequence flow *Order denied* applied.
  - The sequence flow *Shipment received* applied.

- *Change ticket responsible to "Employee"*

  To be executed when:

  - The sequence flow *Approval denied* applied.
  - The sequence flow *Order denied* applied.
  - The sequence flow *Shipment received* applied.

- *Move process ticket into the "Purchasing" queue*

To be executed when the sequence flow *Approved* applied.

- *Move process ticket into the "Post office" queue*

  To be executed when the sequence flow *Order placed* applied.

- *Close ticket successfully*

  To be executed when the sequence flow *Shipment received* applied.

- *Close ticket unsuccessfully*

  To be executed when:

  - The sequence flow *Approval denied* applied.
  - The sequence flow *Order denied* applied.

There are places where the same sequence flow actions should be executed. Therefore it is reasonable to make it possible to link sequence flow actions freely with sequence flows to be able to reuse them.

Create Activities

Click on the *Activities* item in the *Available Process Elements* widget in the left sidebar. This action will expand the *Activities* options and will collapse all others doing an accordion like effect. Click on the *Create New Activity* button.

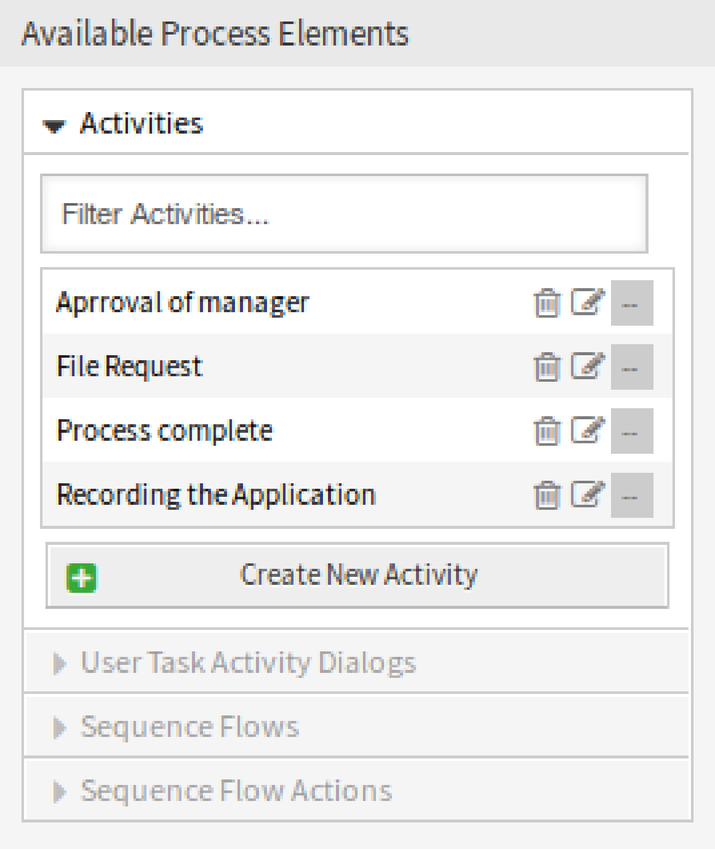
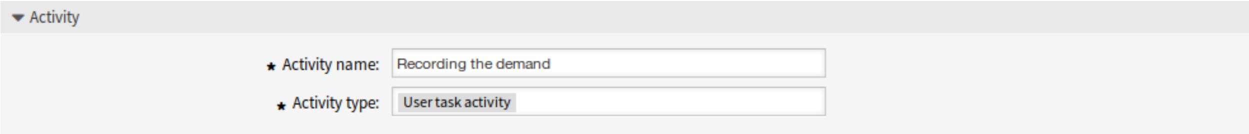


Fig. 75: Book Ordering - Activities

In the opened popup screen fill in the *Activity name* field and select *User task activity* from the *Activity type* drop-down.



▾ Activity

★ Activity name: Recording the demand

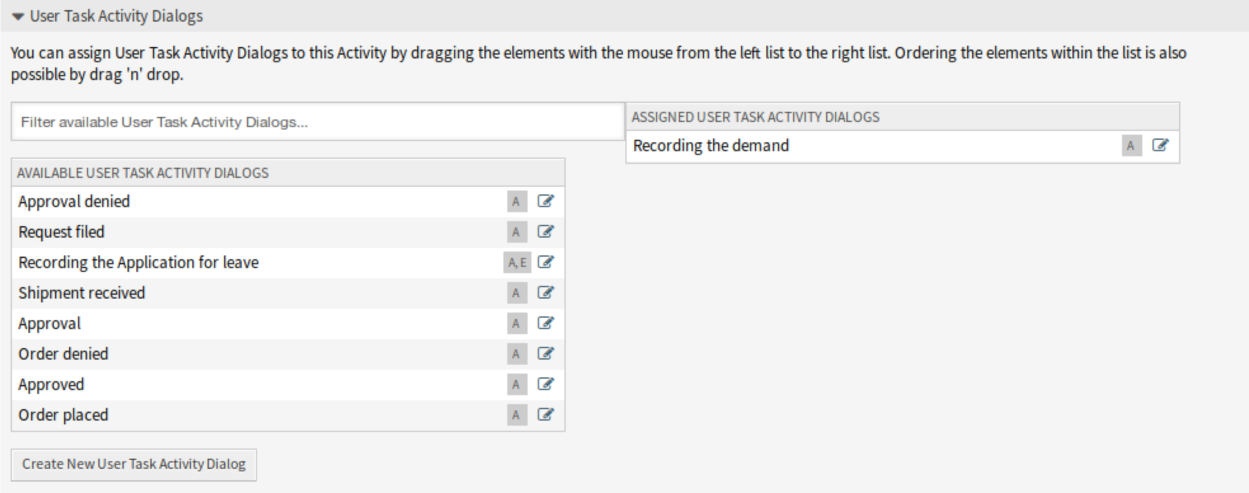
★ Activity type: User task activity

Fig. 76: Book Ordering - Add Activity

To assign dialogs to the activity simple drag the required dialogs from the *Available User Task Activity Dialogs* pool and drop into the *Assigned User Task Activity Dialogs* pool. The order in the *Assigned User Task Activity Dialogs* pool is the order as the dialogs will be presented in the ticket detail view. To modify the order simply drag and drop the dialog within the pool to rearrange it in the correct place.

Note: This order is specially important in the first activity, since the first user task activity dialog for this activity is the only one that is presented when the process starts.

In this example we need to assign only the *Recording the demand* user task activity dialog. Drag this dialog from the *Available User Task Activity Dialogs* pool and drop into the *Assigned User Task Activity Dialogs* pool.



▾ User Task Activity Dialogs

You can assign User Task Activity Dialogs to this Activity by dragging the elements with the mouse from the left list to the right list. Ordering the elements within the list is also possible by drag 'n' drop.

Filter available User Task Activity Dialogs...

AVAILABLE USER TASK ACTIVITY DIALOGS	
Approval denied	A
Request filed	A
Recording the Application for leave	A, E
Shipment received	A
Approval	A
Order denied	A
Approved	A
Order placed	A

ASSIGNED USER TASK ACTIVITY DIALOGS	
Recording the demand	A

Fig. 77: Book Ordering - Assign User Task Activity Dialog

After all fields are filled in, click on the *Save and finish* button to save the changes and go back to the project management screen.

Create the following activities:

- *Recording the demand* (already created before)

  Assign the user task activity dialog *Recording the demand*.

- *Approval*

  Assign the user task activity dialogs *Approval denied* and *Approved*.

- *Order*

  Assign the user task activity dialogs *Order denied* and *Order placed*.

- *Incoming*

  Assign the user task activity dialog *Shipment received*.

- *Process complete*

  This is an activity without possible user task activity dialogs. It will be set after *Approval denied*, *Order denied* or *Shipment received* and represents the end of the process.

Now we can clearly see that activities are precisely defined states of a process ticket. After a successful sequence flow a process ticket moves from one activity to another.

Create Process Path

Let us conclude our example with the last missing piece in the puzzle, the process as the a flow describer. In our case this is the whole ordering workflow. Other processes could be office supply ordering or completely different processes.

The process has a starting point which consists of the start activity and the start user task activity dialog. For any new book order, the first user task activity dialog of the first activity is the first screen that is displayed. If this is completed and saved, the process ticket will be created and can follow the configured workflow.

The process also contains the directions for how the process ticket can move through the process. Let's call this *process path*. It consists of the start activity, one or more sequence flows (possibly with sequence flow actions) and other activities.

Assuming that the activities has already assigned their user task activity dialogs, drag an activity from the accordion in the *Available Process Elements* widget in the left sidebar and drop it into the canvas area below the process information. Notice that an arrow from the process start (white circle) to the activity is placed automatically. This is the first activity and its first user task activity dialog is the first screen that will be shown when the process starts.

Next, drag another activity into the canvas too. Now we will have two activities in the canvas. The first one is connected to the start point and the second has no connections. You can hover the mouse over each activity to reveal their own activity dialogs.

Then let's create the process path (connection) between this two activities. For this we will use the sequence flows. Click on sequence flow in the accordion, drag a sequence flow and drop it inside the first activity. As soon as the sequence flow is dropped the end point of the sequence flow arrow will be placed next to the process start point. Drag the sequence flow arrow end point and drop it inside the other activity to create the connection between the activities.

Now that the process path between the actions is defined, then we need to assign the sequence flow actions to the sequence flow. Double click the sequence flow label in the canvas to open a new popup window.

After the sequence flow actions are assigned, click on the *Save* button to go back to the main process edit screen. Click on *Save* button below the canvas to save all other changes.

Complete the process path by adding the following activities, sequence flows and sequence flow actions:

- *Recording the demand* (already created before)

  Possible sequence flow: *Approval*

  Starting activity: *Recording the demand*

  Next activity: *Approval*

  If the condition of this activity is fulfilled, the ticket will move to activity *Approval*.

  Additionally, the following sequence flow actions are executed:

  - *Move the process ticket into the "Management" queue*

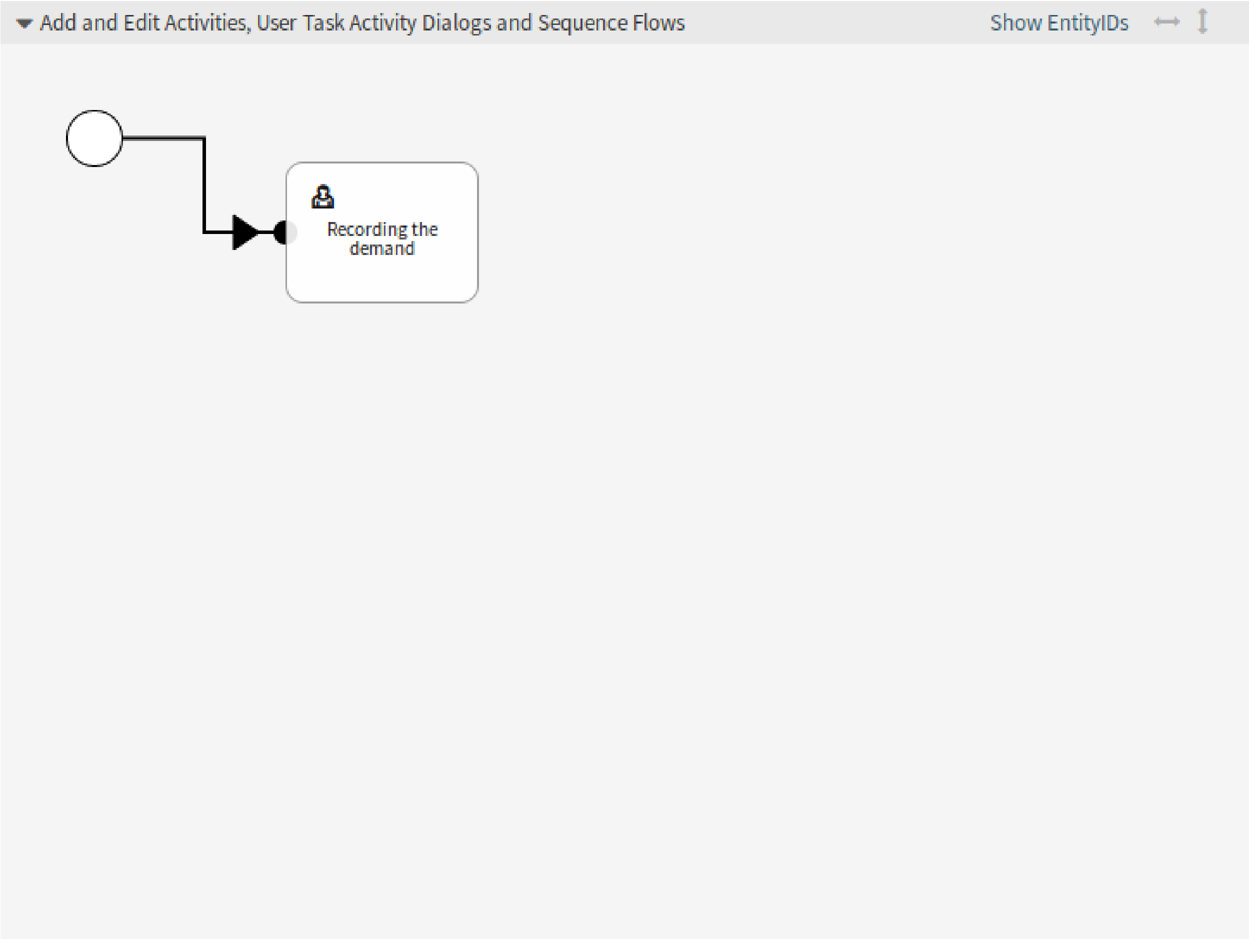

Add and Edit Activities, User Task Activity Dialogs and Sequence Flows
Show EntityIDs
Recording the demand

Fig. 78: Book Ordering - First Activity On Canvas

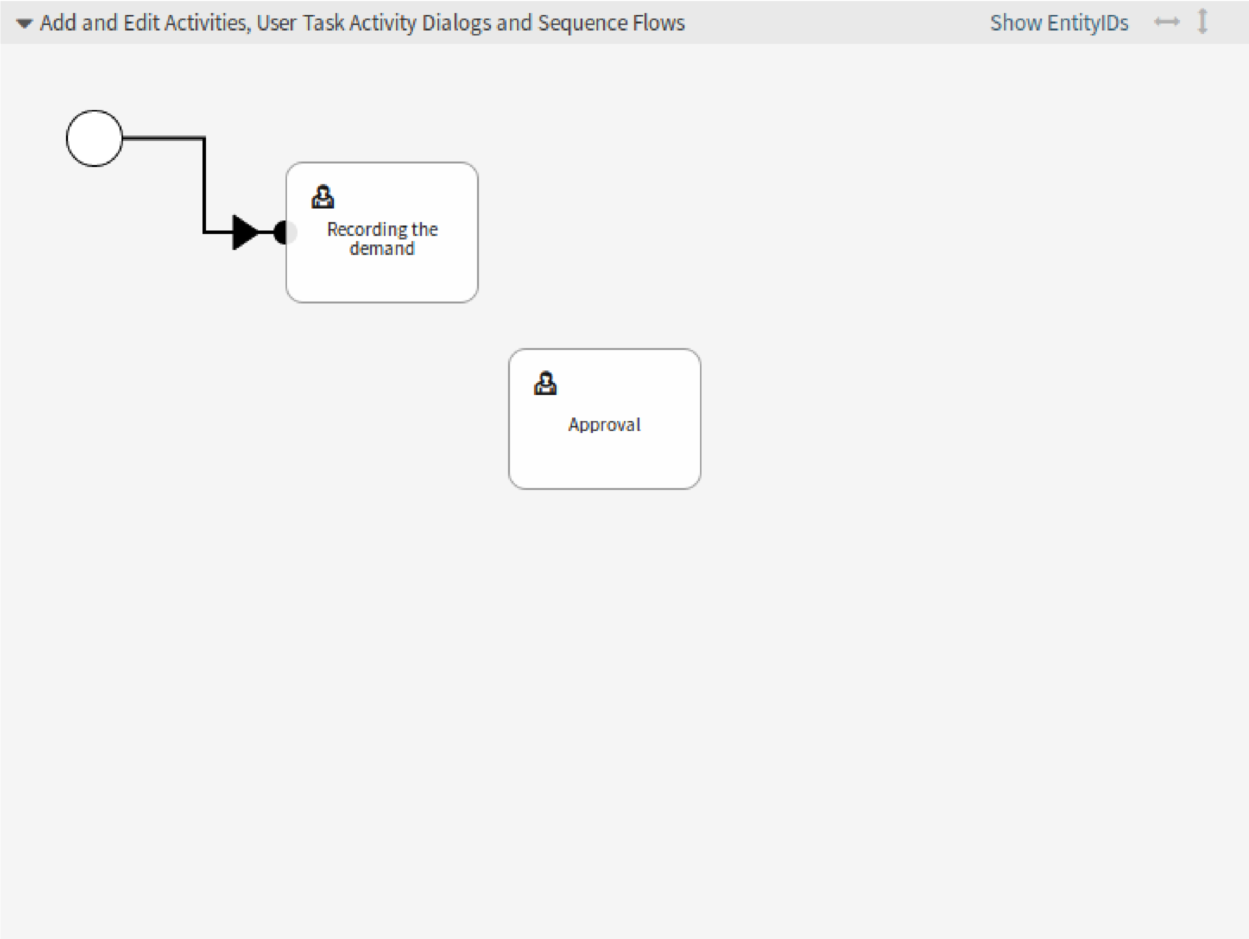

Add and Edit Activities, User Task Activity Dialogs and Sequence Flows
Show EntityIDs
Recording the demand
Approval

Fig. 79: Book Ordering - Second Activity On Canvas

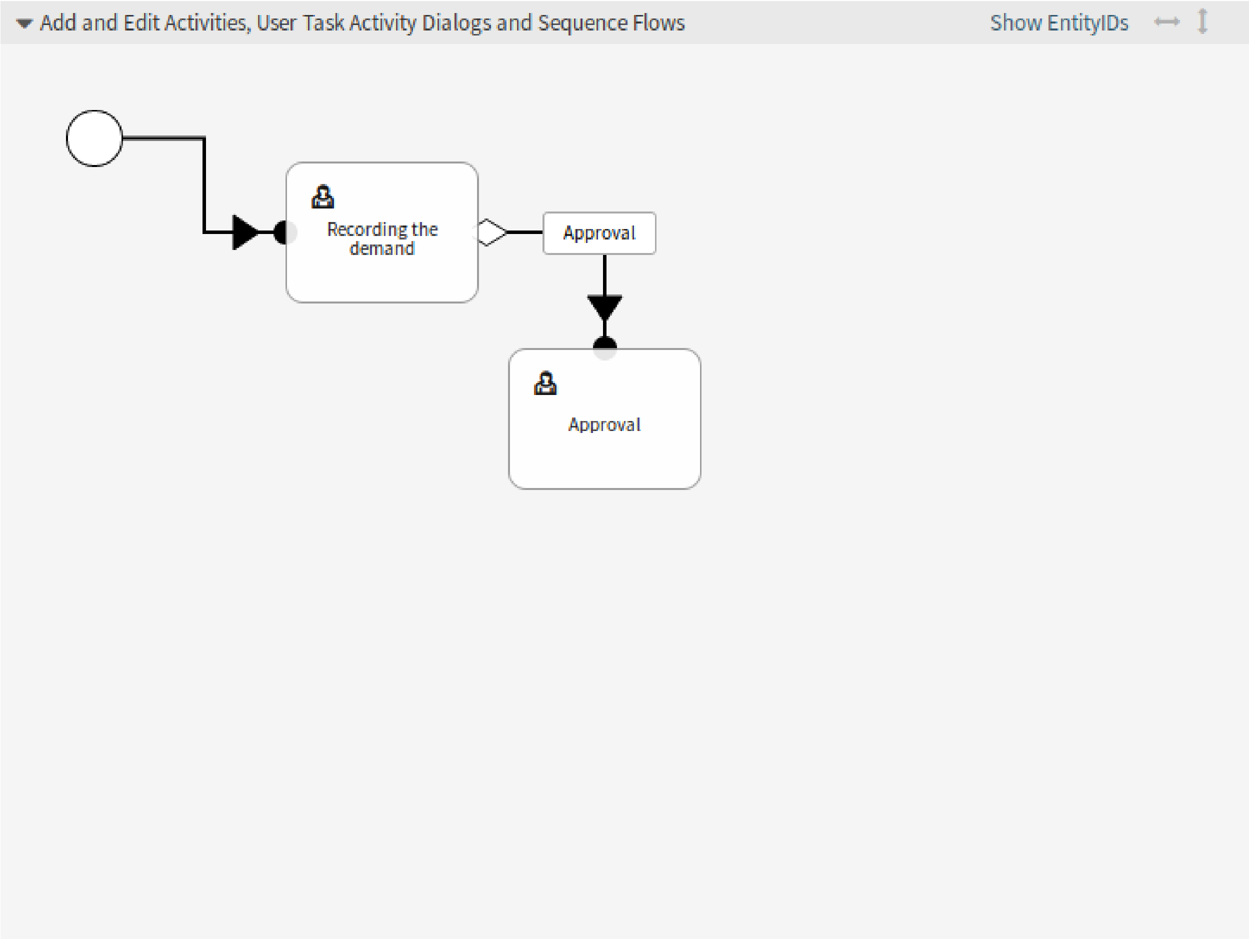
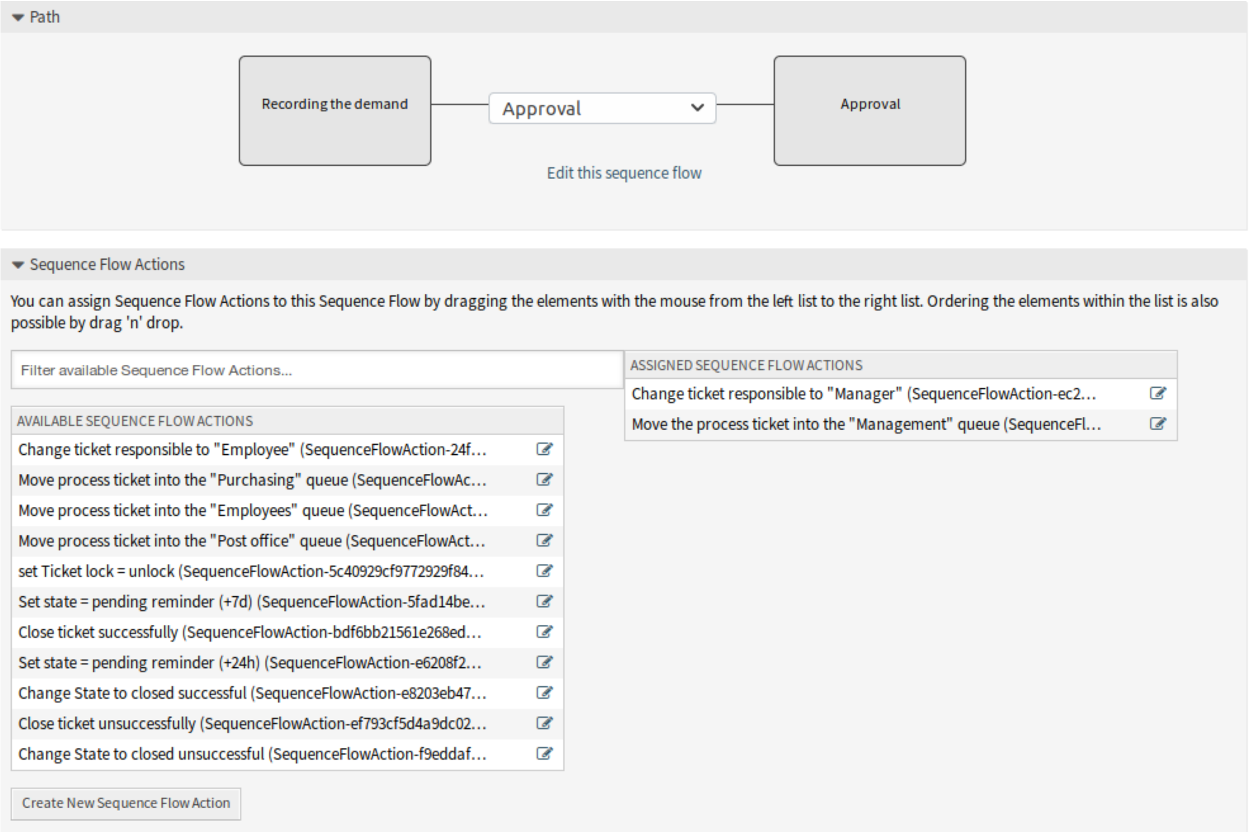

Add and Edit Activities, User Task Activity Dialogs and Sequence Flows
Show EntityIDs
Recording the demand
Approval
Approval

Fig. 80: Book Ordering - First Sequence Flow On Canvas



You can assign Sequence Flow Actions to this Sequence Flow by dragging the elements with the mouse from the left list to the right list. Ordering the elements within the list is also possible by drag 'n' drop.

AVAILABLE SEQUENCE FLOW ACTIONS

- Change ticket responsible to "Employee" (SequenceFlowAction-24f…
- Move process ticket into the "Purchasing" queue (SequenceFlowAc…
- Move process ticket into the "Employees" queue (SequenceFlowAct…
- Move process ticket into the "Post office" queue (SequenceFlowAct…
- set Ticket lock = unlock (SequenceFlowAction-5c40929cf9772929f84…
- Set state = pending reminder (+7d) (SequenceFlowAction-5fad14be…
- Close ticket successfully (SequenceFlowAction-bdf6bb21561e268ed…
- Set state = pending reminder (+24h) (SequenceFlowAction-e6208f2…
- Change State to closed successful (SequenceFlowAction-e8203eb47…
- Close ticket unsuccessfully (SequenceFlowAction-ef793cf5d4a9dc02…
- Change State to closed unsuccessful (SequenceFlowAction-f9eddaf…

ASSIGNED SEQUENCE FLOW ACTIONS

- Change ticket responsible to "Manager" (SequenceFlowAction-ec2…
- Move the process ticket into the "Management" queue (SequenceFl…

Fig. 81: Book Ordering - Assign First Sequence Flow Action

  - *Change ticket responsible to "Manager"*

  The activity *Recording the demand* is a defined step of the process ticket, where there is the possibility for the sequence flow *Approval*. If this applies, the ticket will move to the next activity *Approval*, and the sequence flow actions *Move the process ticket into the "Management" queue* and *Change ticket responsible to "Manager"* are executed. In the activity *Approval*, the user task activity dialogs *Approval denied* and *Approved* are available.

- *Approval*

  Possible sequence flow: *Approval denied*

  Starting activity: *Approval*

  Next activity: *Process complete*

  If this matches, the process ticket will move to activity *Process complete*.

  Additionally, the following sequence flow actions are executed:

  - *Move process ticket into the "Employees" queue*
  - *Change ticket responsible to "Employee"*
  - *Close ticket unsuccessfully*

  Possible sequence flow: *Approved*

  Starting activity: *Approval*

  Next activity: *Order*

  If this matches, the process ticket will move to activity *Order*.

  Additionally, the following sequence flow actions are executed:

  - *Move process ticket into the "Purchasing" queue*

  We can see that from the current activity, which defines a step of the process ticket, there are one or more possibilities for sequence flow which have exactly one target activity (and possibly one or more sequence flow actions).

- *Order*

  Possible sequence flow: *Order denied*

  Starting activity: *Order*

  Next activity: *Process complete*

  If this matches, the process ticket will move to activity *Process complete*.

  Additionally, the following sequence flow actions are executed:

  - *Move process ticket into the "Employees" queue*
  - *Change ticket responsible to "Employee"*
  - *Close ticket unsuccessfully*

  Possible sequence flow: *Order placed*

  Starting activity: *Order*

  Next activity: *Incoming*

  If this matches, the process ticket will move to activity *Incoming*.

  Additionally, the following sequence flow actions are executed:

  - *Move process ticket into the "Post office" queue*

- *Incoming*

  Possible sequence flow: *Shipment received*

  Starting activity: *Incoming*

  Next activity: *Process complete*

  If this matches, the process ticket will move to activity *Process complete*.

  Additionally, the following sequence flow actions are executed:

  - *Move process ticket into the "Employees" queue*
  - *Change ticket responsible to "Employee"*
  - *Close ticket successfully*

The complete process path for the book ordering process will then look like this:

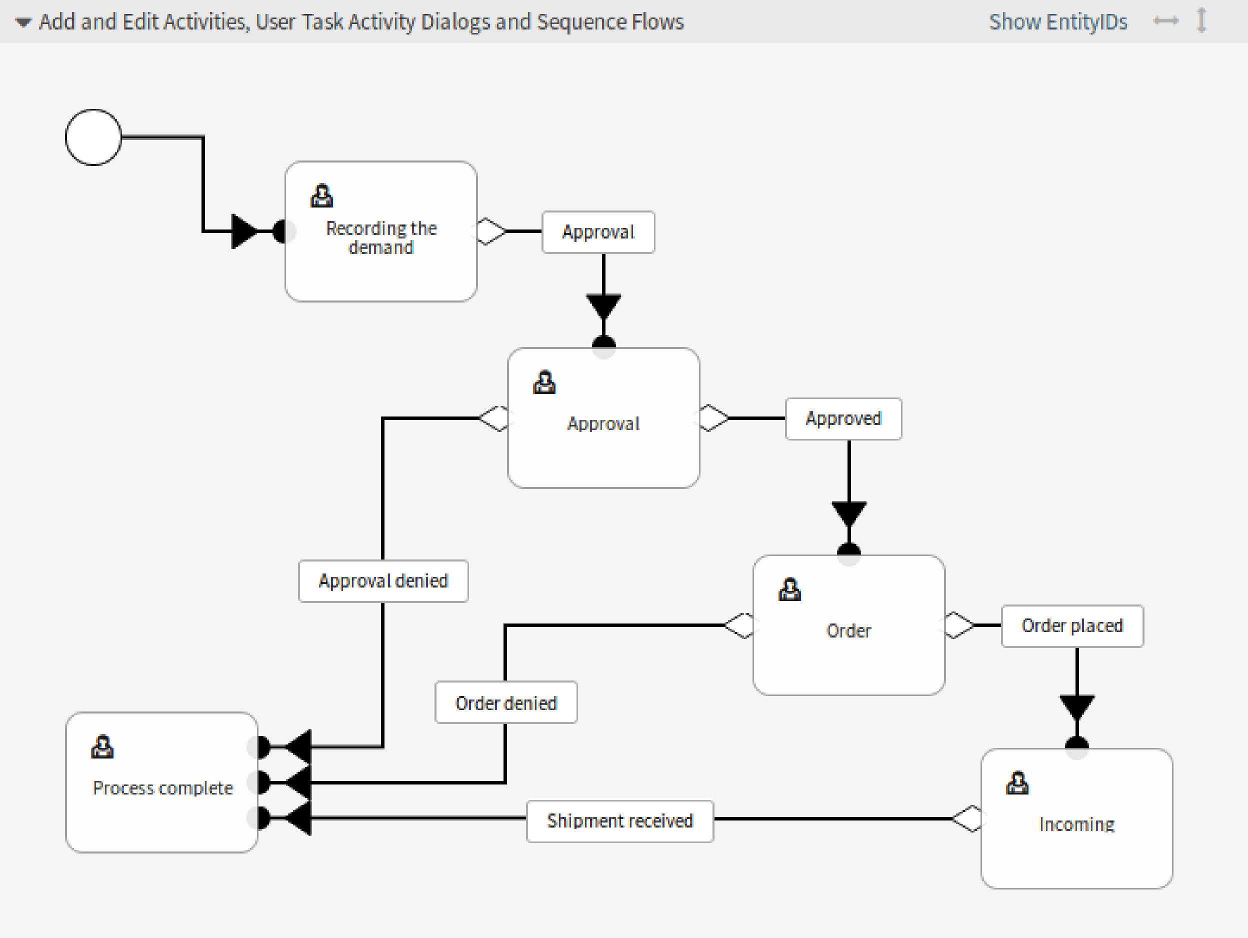


Fig. 82: Book Ordering - Process Complete

After you finish the process path, click on *Save and finish* button below the canvas to go back to the process management screen.

Click on the *Deploy All Processes* button in the left sidebar. This will gather all processes information form the database and create a cache file (in Perl language). This cache file is actually the processes configuration that the system will use to create or use process tickets.

Note: Any change that is made on the process will require to re-deploy the process in order to get the change reflected in the system.

5.5 Web Services

In a connected world, a ticket system needs to be able to react to requests from other systems and also to send requests or information to other systems:

- CRM systems
- Project management systems
- Documentation management systems
- and many more

The ticket system must be reachable by other services without manual intervention by an agent.

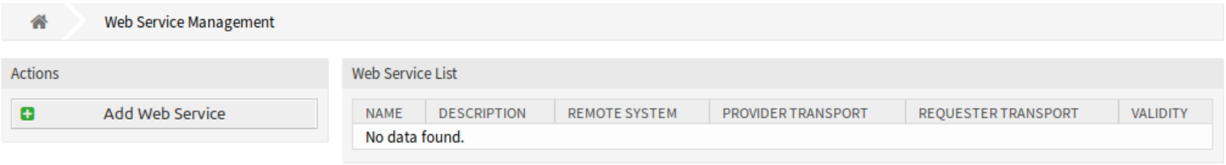
OTRS supports this requirement by the *Generic Interface*. It empowers the administrator to create a web service for a specific task without scripting language knowledge. OTRS reacts on incoming REST or SOAP requests and create objects or provides object data to other systems transparently.

A web service is a communication method between two systems, in our case OTRS and a remote system. In its configuration, the *operation* or *invoker* determine the direction of communication, and the *mapping* and *transport* take care of how the data is received and interpreted.

In its configuration it can be defined what actions the web service can perform internally (operation), what actions the OTRS request can perform remote system (invokers), how data is converted from one system to the other (mapping), and over which protocol the communication will take place (transport).

The generic interface is the framework that makes it possible to create web services for OTRS in a predefined way, using already made building blocks that are independent from each other and interchangeable.

Use this screen to manage web services in the system. A fresh OTRS installation contains no web service by default. The web service management screen is available in the *Web Services* module of the *Processes & Automation* group.



NAME	DESCRIPTION	REMOTE SYSTEM	PROVIDER TRANSPORT	REQUESTER TRANSPORT	VALIDITY
No data found.					

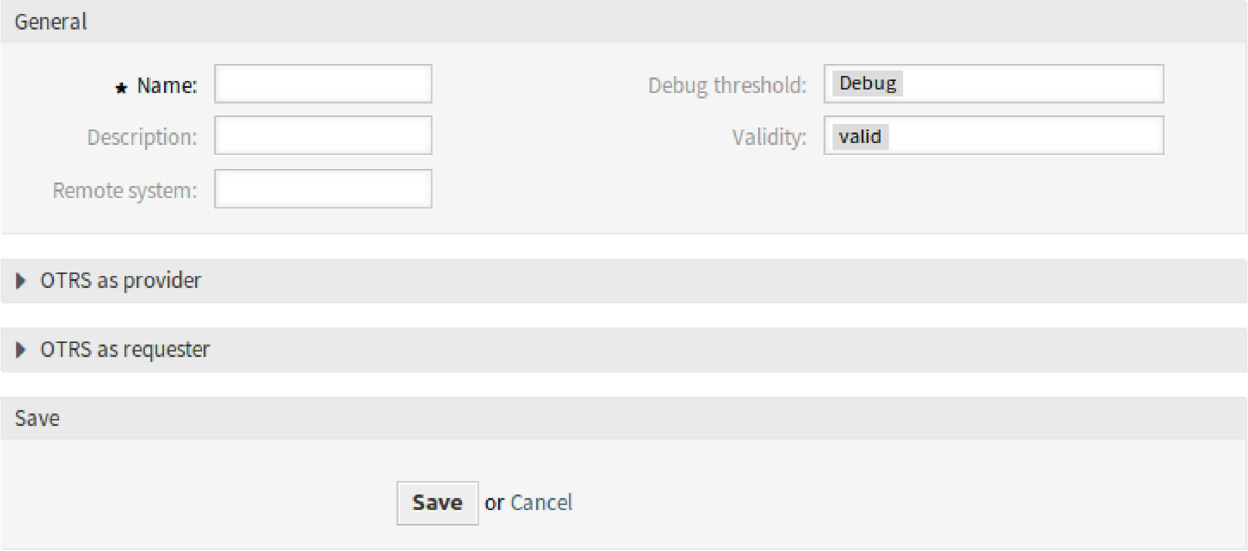
Fig. 83: Web Service Management Screen

5.5.1 Manage Web Services

To create a web service:

1. Click on the *Add Web Service* button in the left sidebar.
2. Fill in the required fields.

3. Click on the *Save* button.



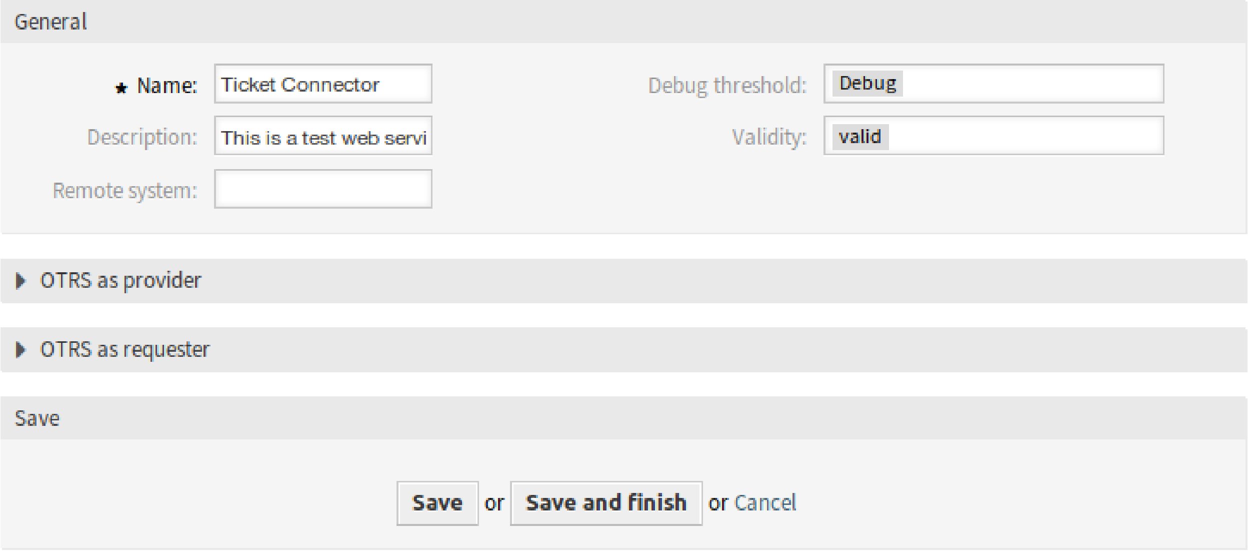
Debug threshold: Debug

Validity: valid

Fig. 84: Create New Web Service Screen

To edit a web service:

1. Click on a web service in the list of web services.
2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.



★ Name: Ticket Connector

Description: This is a test web servi

Debug threshold: Debug

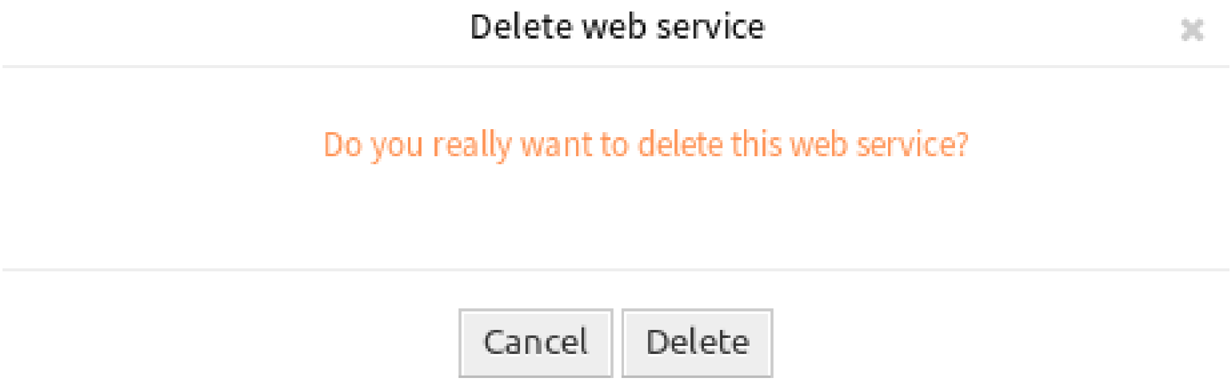
Validity: valid

Fig. 85: Edit Web Service Screen

To delete a web service:

1. Click on a web service in the list of web services.
2. Click on the *Delete Web Service* button in the left sidebar.

3. Click on the *Delete* button in the confirmation dialog.



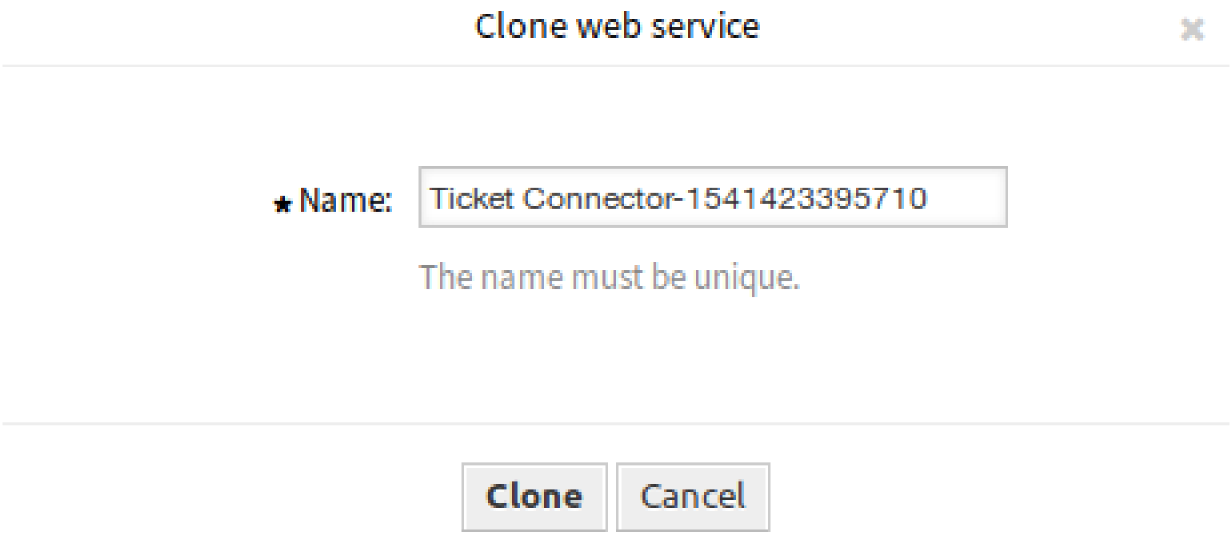
Delete web service

Do you really want to delete this web service?

Fig. 86: Delete Web Service Screen

To clone a web service:

1. Click on a web service in the list of web services.
2. Click on the *Clone Web Service* button in the left sidebar.
3. Enter a new name for the web service.



Clone web service

★ Name: Ticket Connector-1541423395710

The name must be unique.

Fig. 87: Clone Web Service Screen

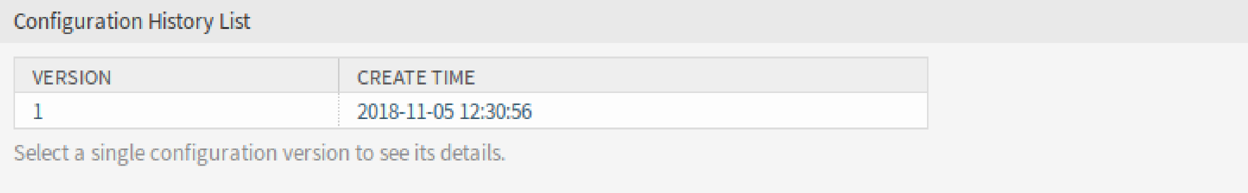
To export a web service:

1. Click on a web service in the list of web services.
2. Click on the *Export Web Service* button in the left sidebar.
3. Choose a location in your computer to save the `Export_ACL.yml` file.

Warning: All stored passwords in the web service configuration will be exported in plain text format.

To see the configuration history of a web service:

1. Click on a web service in the list of web services.
2. Click on the *Configuration History* button in the left sidebar.



Configuration History List

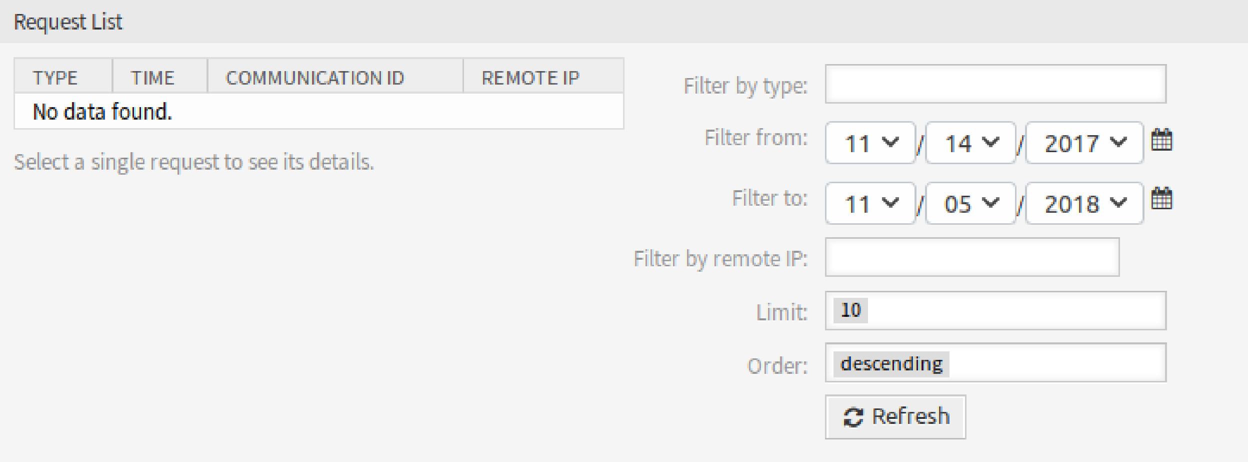
VERSION	CREATE TIME
1	2018-11-05 12:30:56

Select a single configuration version to see its details.

Fig. 88: Web Service Configuration History Screen

To use the debugger for a web service:

1. Click on a web service in the list of web services.
2. Click on the *Debugger* button in the left sidebar.



Request List

TYPE	TIME	COMMUNICATION ID	REMOTE IP
No data found.			

Select a single request to see its details.

Filter by type:

Filter from: 11 / 14 / 2017

Filter to: 11 / 05 / 2018

Filter by remote IP:

Limit: 10

Order: descending

Fig. 89: Web Service Debugger Screen

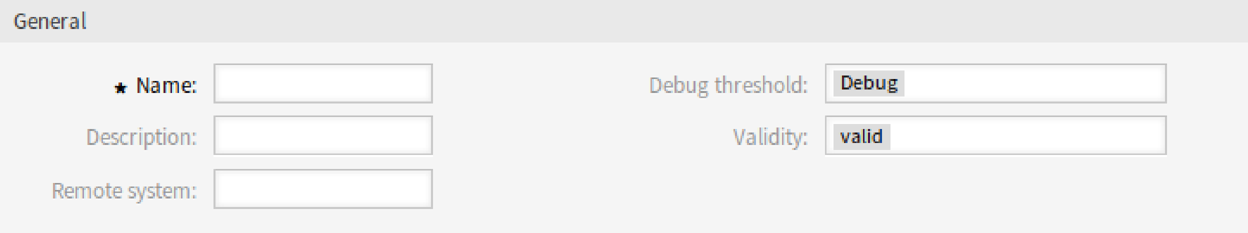
To import a web service:

1. Click on the *Add Web Service* button in the left sidebar.
2. Click on the *Import Web Service* button in the left sidebar.
3. Click on the *Browse…* button in the dialog.
4. Select a previously exported `.yml` file.
5. Add a name for the imported web service (optional). Otherwise the name will be taken from the configuration file name.
6. Click on the *Import* button.

5.5.2 Web Service Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

General Web Service Settings



General

★ Name:

Description:

Remote system:

Debug threshold: Debug

Validity: valid

Fig. 90: Web Service Settings - General

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Description Like comment, but longer text can be added here.

Remote system

Debug threshold The default value is *Debug*. When configured in this manner all communication logs are registered in the database. Each subsequent debug threshold value is more restrictive and discards communication logs of lower order than the one set in the system.

Debug threshold levels (from lower to upper):

- Debug
- Info
- Notice
- Error

Validity Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.

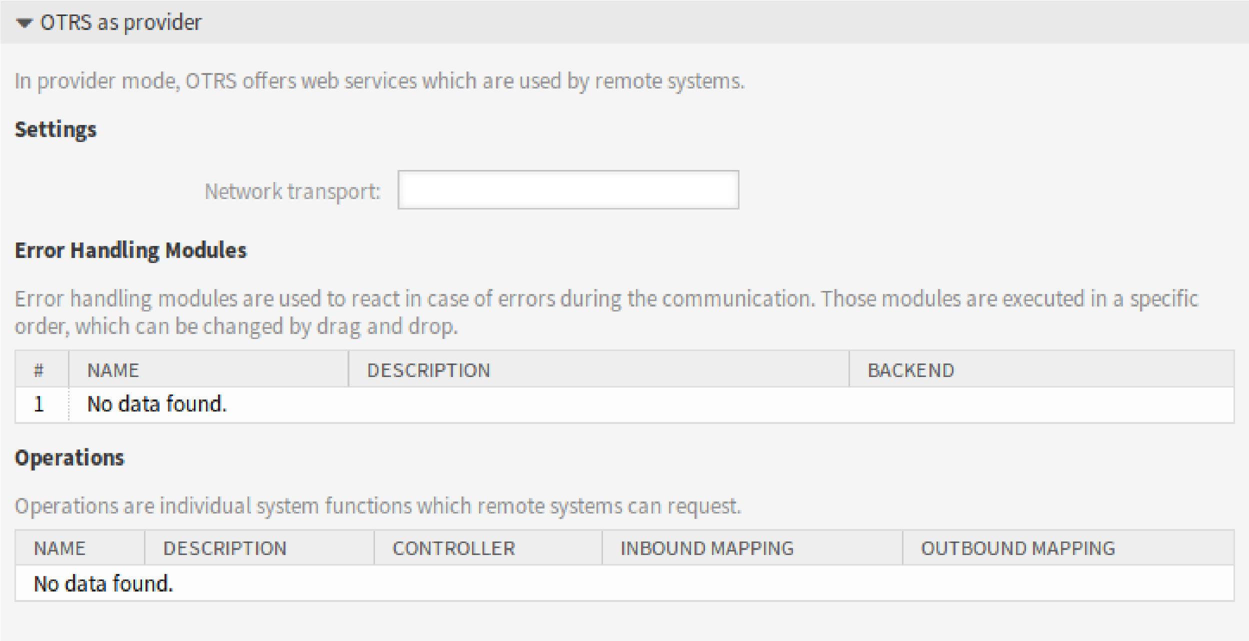
Provider Web Service Settings

Network transport Select which network transport would you like to use with the web service. Possible values are *HTTP::REST* and *HTTP::SOAP*.

> **Note:** After selecting the transport method, you have to save the configuration with clicking on the *Save* button. A *Configuration* button will be displayed next to this field.

Configuration The *Configuration* button is visible only, after a network transport was selected and saved. See the configuration for *OTRS as Provider - HTTP::REST* and *OTRS as Provider - HTTP::SOAP* below.

Add Operation This option is visible only, after a network transport was selected and saved. Selecting an operation will open a new screen for its configuration.



▼ OTRS as provider

In provider mode, OTRS offers web services which are used by remote systems.

Settings

Network transport:

Error Handling Modules

Error handling modules are used to react in case of errors during the communication. Those modules are executed in a specific order, which can be changed by drag and drop.

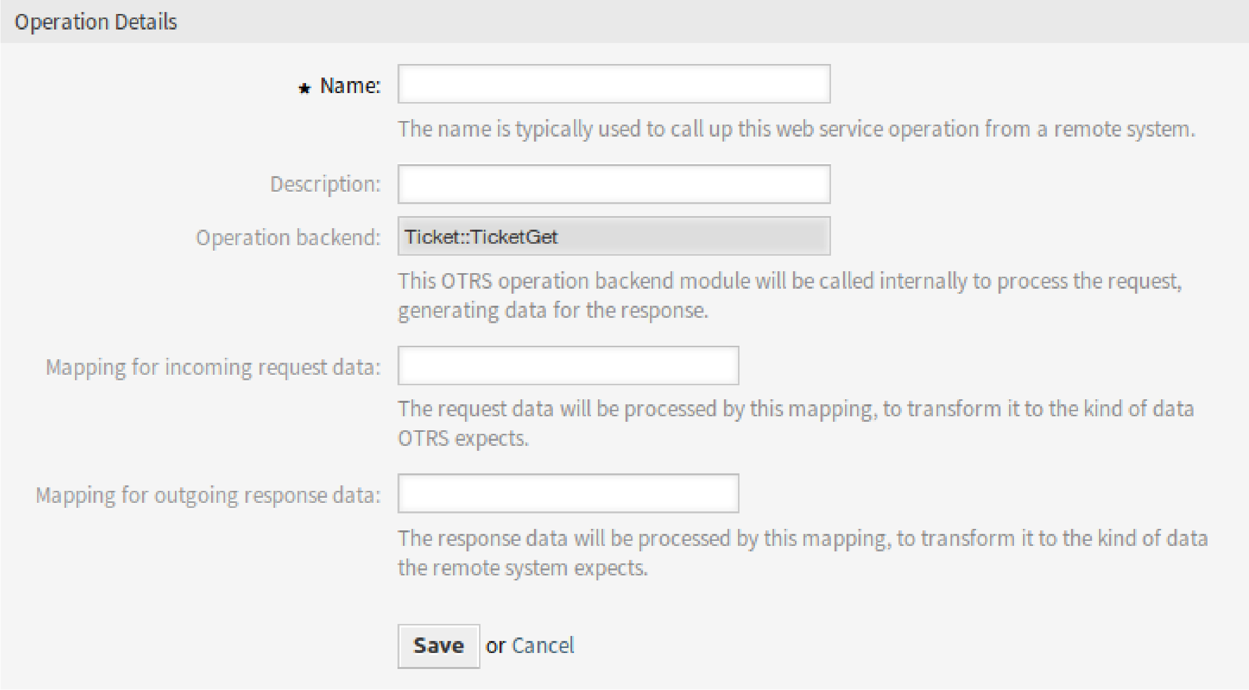
#	NAME	DESCRIPTION	BACKEND
1	No data found.		

Operations

Operations are individual system functions which remote systems can request.

NAME	DESCRIPTION	CONTROLLER	INBOUND MAPPING	OUTBOUND MAPPING
No data found.				

Fig. 91: Web Service Settings - OTRS as Provider



Operation Details

★ Name:

The name is typically used to call up this web service operation from a remote system.

Description:

Operation backend: Ticket::TicketGet

This OTRS operation backend module will be called internally to process the request, generating data for the response.

Mapping for incoming request data:

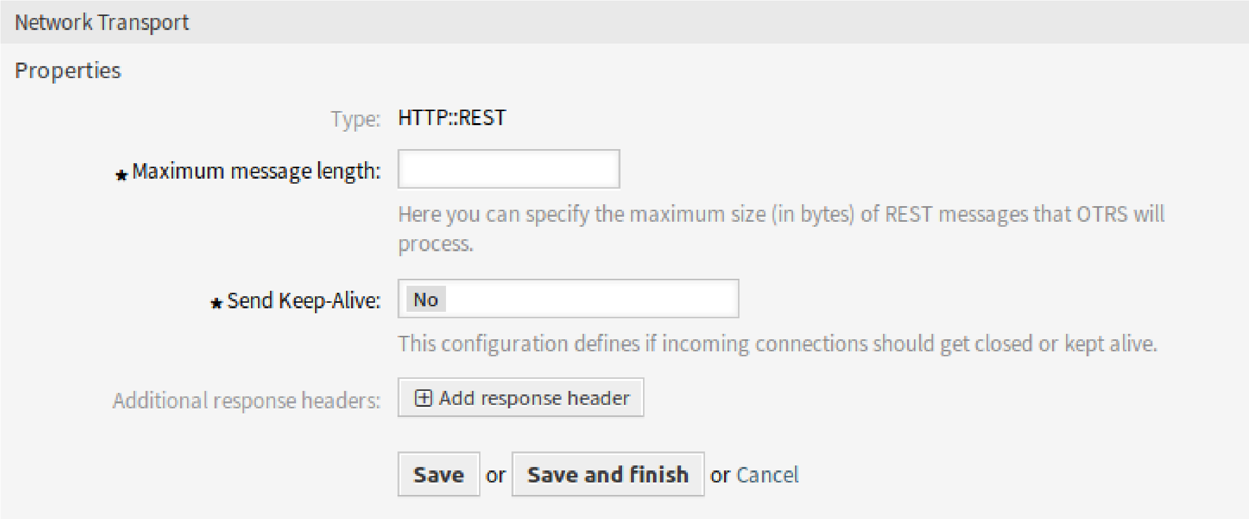
The request data will be processed by this mapping, to transform it to the kind of data OTRS expects.

Mapping for outgoing response data:

The response data will be processed by this mapping, to transform it to the kind of data the remote system expects.

Fig. 92: Web Service Settings - OTRS as Provider - Operation

OTRS as Provider - HTTP::REST



Network Transport

Properties

Type: HTTP::REST

* Maximum message length:

Here you can specify the maximum size (in bytes) of REST messages that OTRS will process.

* Send Keep-Alive: No

This configuration defines if incoming connections should get closed or kept alive.

Fig. 93: Web Service Settings - OTRS as Provider - HTTP::REST

OTRS as Provider - HTTP::SOAP

Requester Web Service Settings

Network transport Select which network transport would you like to use with the web service. Possible values are *HTTP::REST* and *HTTP::SOAP*.

> **Note:** After selecting the transport method, you have to save the configuration with clicking on the *Save* button. A *Configuration* button will be displayed next to this field.

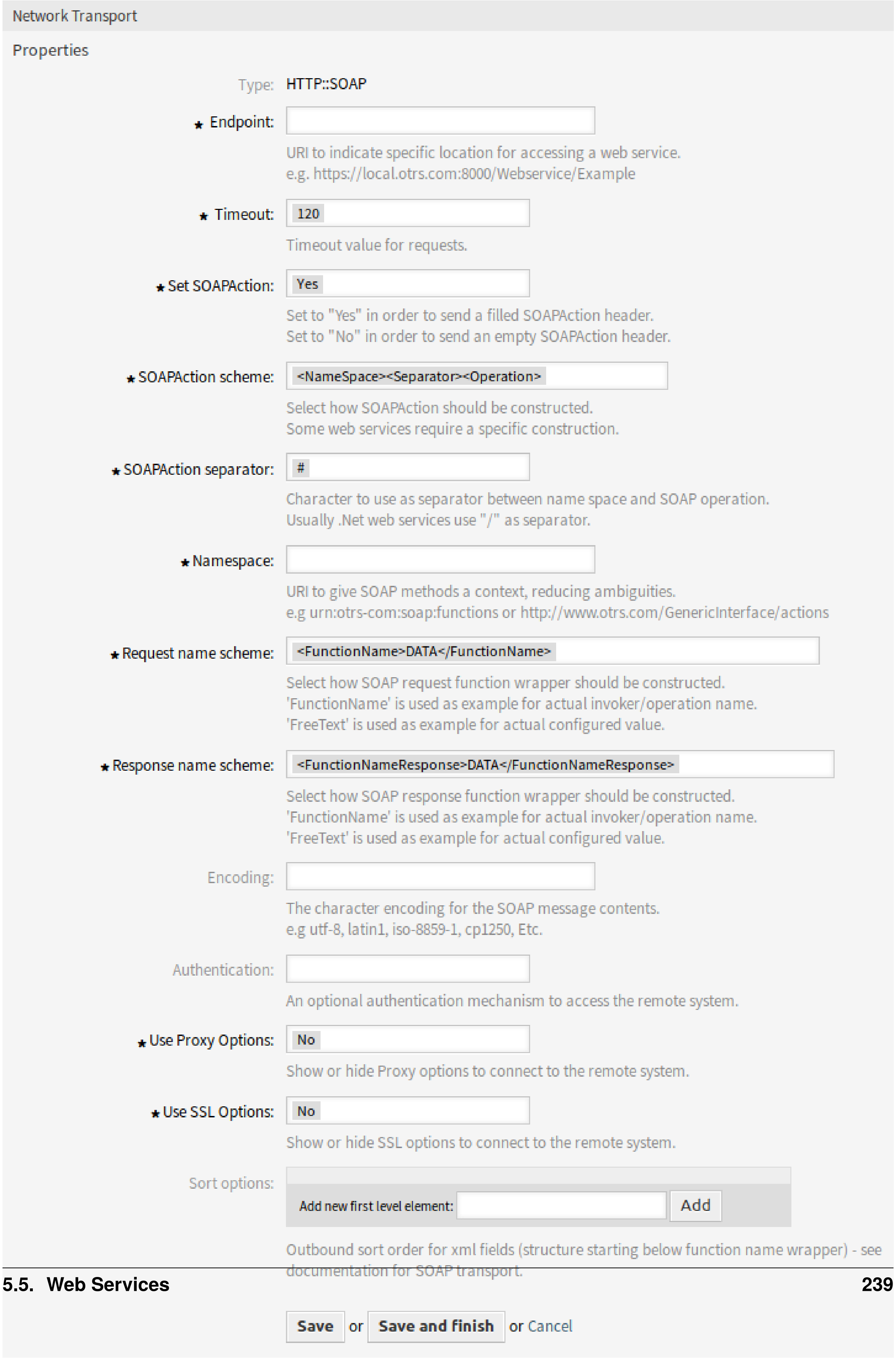
Configuration The *Configuration* button is visible only, after a network transport was selected and saved. See the configuration for *OTRS as Requester - HTTP::REST* and *OTRS as Requester - HTTP::SOAP* below.

It is possible to use both object and array format as a JSON response of the remote system. However, in the case it is an array, system stores it as an object internally, where `ArrayData` is used as a key and a value is an array. Because of that, responded JSON array can be mapped efficiently, but has to be considered as an object described above (key is `ArrayData`, but * can also be used as wildcard).

Add error handling module This option is visible only, after a network transport was selected and saved. Selecting an operation will open a new screen for its configuration.

OTRS as Requester - HTTP::REST

OTRS as Requester - HTTP::SOAP



Network Transport

Properties

Type: HTTP::SOAP

★ Endpoint:

URI to indicate specific location for accessing a web service.
e.g. https://local.otrs.com:8000/Webservice/Example

★ Timeout: 120

Timeout value for requests.

★ Set SOAPAction: Yes

Set to "Yes" in order to send a filled SOAPAction header.
Set to "No" in order to send an empty SOAPAction header.

★ SOAPAction scheme: <NameSpace><Separator><Operation>

Select how SOAPAction should be constructed.
Some web services require a specific construction.

★ SOAPAction separator: #

Character to use as separator between name space and SOAP operation.
Usually .Net web services use "/" as separator.

★ Namespace:

URI to give SOAP methods a context, reducing ambiguities.
e.g urn:otrs-com:soap:functions or http://www.otrs.com/GenericInterface/actions

★ Request name scheme: <FunctionName>DATA</FunctionName>

Select how SOAP request function wrapper should be constructed.
'FunctionName' is used as example for actual invoker/operation name.
'FreeText' is used as example for actual configured value.

★ Response name scheme: <FunctionNameResponse>DATA</FunctionNameResponse>

Select how SOAP response function wrapper should be constructed.
'FunctionName' is used as example for actual invoker/operation name.
'FreeText' is used as example for actual configured value.

Encoding:

The character encoding for the SOAP message contents.
e.g utf-8, latin1, iso-8859-1, cp1250, Etc.

Authentication:

An optional authentication mechanism to access the remote system.

★ Use Proxy Options: No

Show or hide Proxy options to connect to the remote system.

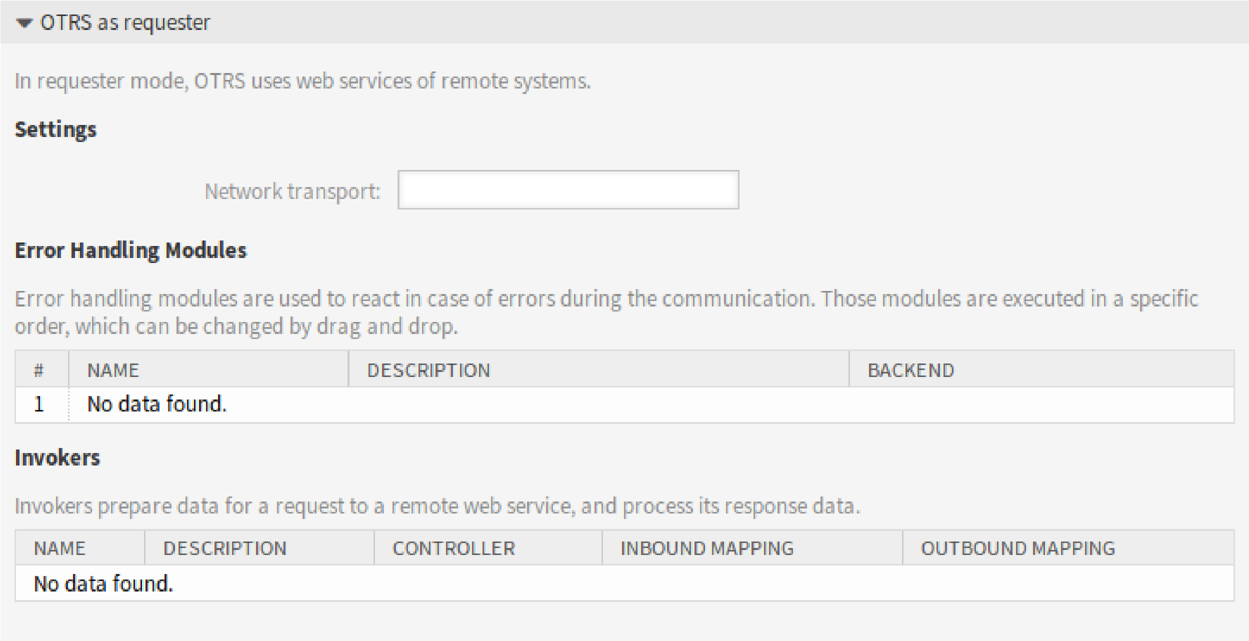
★ Use SSL Options: No

Show or hide SSL options to connect to the remote system.

Sort options:

Add new first level element:

Outbound sort order for xml fields (structure starting below function name wrapper) - see documentation for SOAP transport.



In requester mode, OTRS uses web services of remote systems.

Settings

Network transport:

Error Handling Modules

Error handling modules are used to react in case of errors during the communication. Those modules are executed in a specific order, which can be changed by drag and drop.

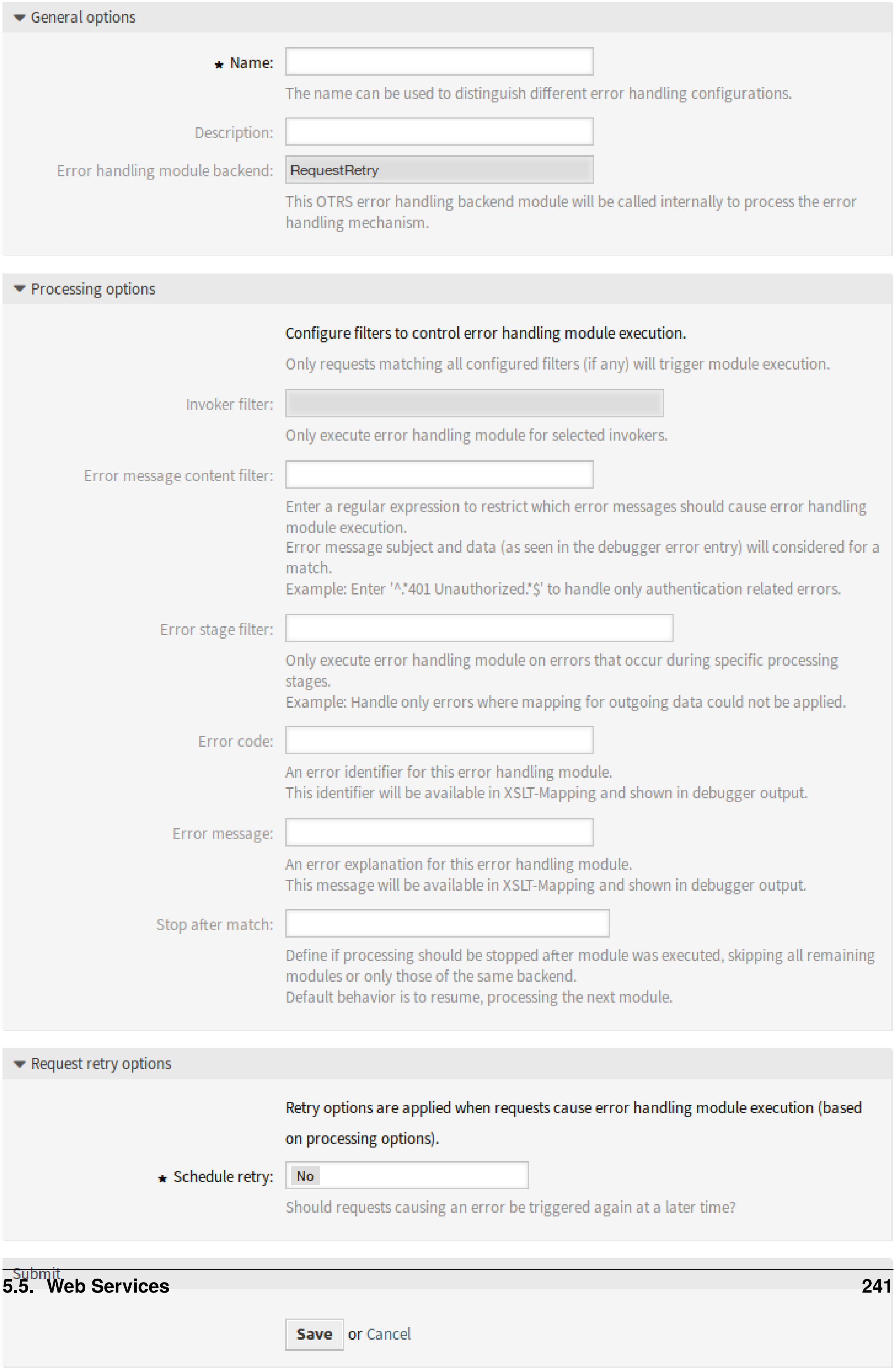
#	NAME	DESCRIPTION	BACKEND
1	No data found.		

Invokers

Invokers prepare data for a request to a remote web service, and process its response data.

NAME	DESCRIPTION	CONTROLLER	INBOUND MAPPING	OUTBOUND MAPPING
No data found.				

Fig. 95: Web Service Settings - OTRS as Requester



▾ General options

★ Name:

The name can be used to distinguish different error handling configurations.

Description:

Error handling module backend: RequestRetry

This OTRS error handling backend module will be called internally to process the error handling mechanism.

▾ Processing options

Configure filters to control error handling module execution.

Only requests matching all configured filters (if any) will trigger module execution.

Invoker filter:

Only execute error handling module for selected invokers.

Error message content filter:

Enter a regular expression to restrict which error messages should cause error handling module execution.
Error message subject and data (as seen in the debugger error entry) will considered for a match.
Example: Enter '^.*401 Unauthorized.*$' to handle only authentication related errors.

Error stage filter:

Only execute error handling module on errors that occur during specific processing stages.
Example: Handle only errors where mapping for outgoing data could not be applied.

Error code:

An error identifier for this error handling module.
This identifier will be available in XSLT-Mapping and shown in debugger output.

Error message:

An error explanation for this error handling module.
This message will be available in XSLT-Mapping and shown in debugger output.

Stop after match:

Define if processing should be stopped after module was executed, skipping all remaining modules or only those of the same backend.
Default behavior is to resume, processing the next module.

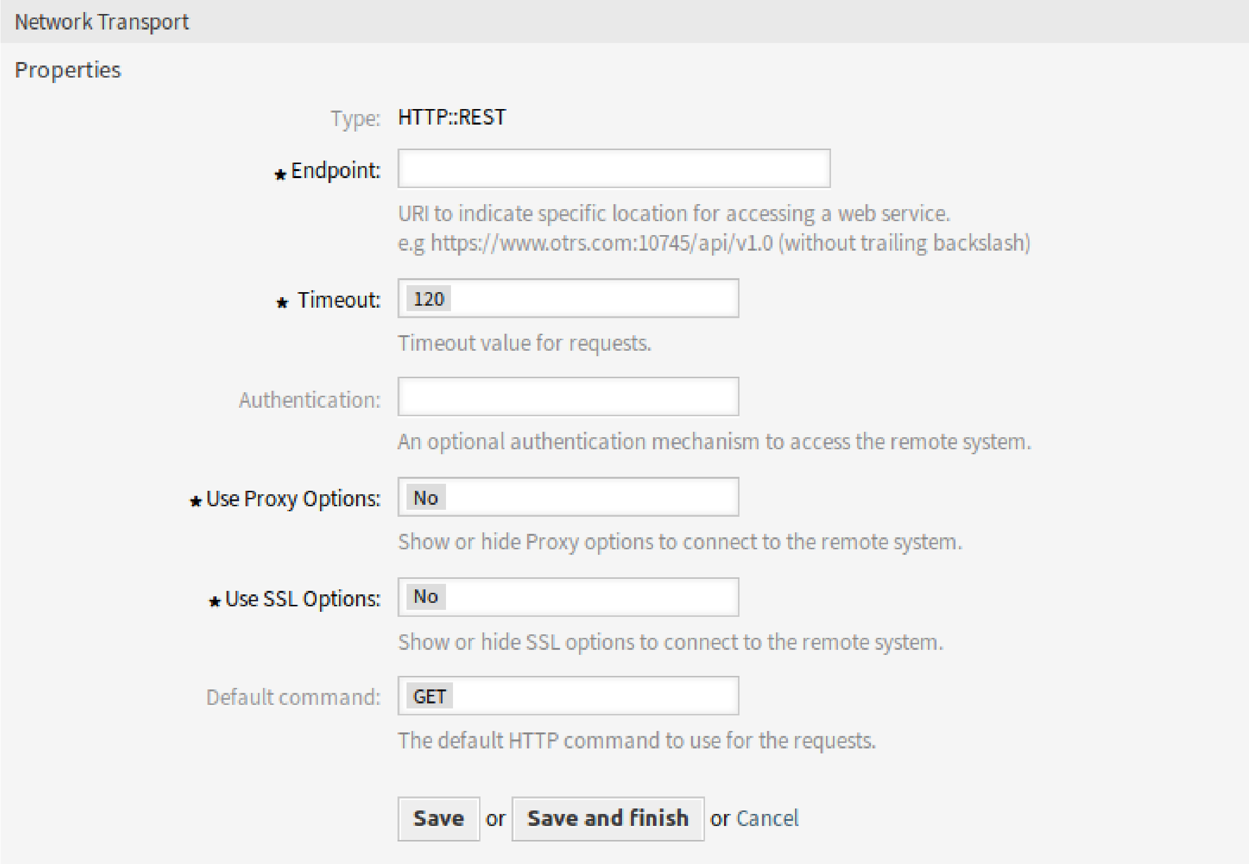
▾ Request retry options

Retry options are applied when requests cause error handling module execution (based on processing options).

★ Schedule retry: No

Should requests causing an error be triggered again at a later time?

Submit



Network Transport

Properties

Type: HTTP::REST

★ Endpoint:

URI to indicate specific location for accessing a web service.
e.g https://www.otrs.com:10745/api/v1.0 (without trailing backslash)

★ Timeout: 120

Timeout value for requests.

Authentication:

An optional authentication mechanism to access the remote system.

★ Use Proxy Options: No

Show or hide Proxy options to connect to the remote system.

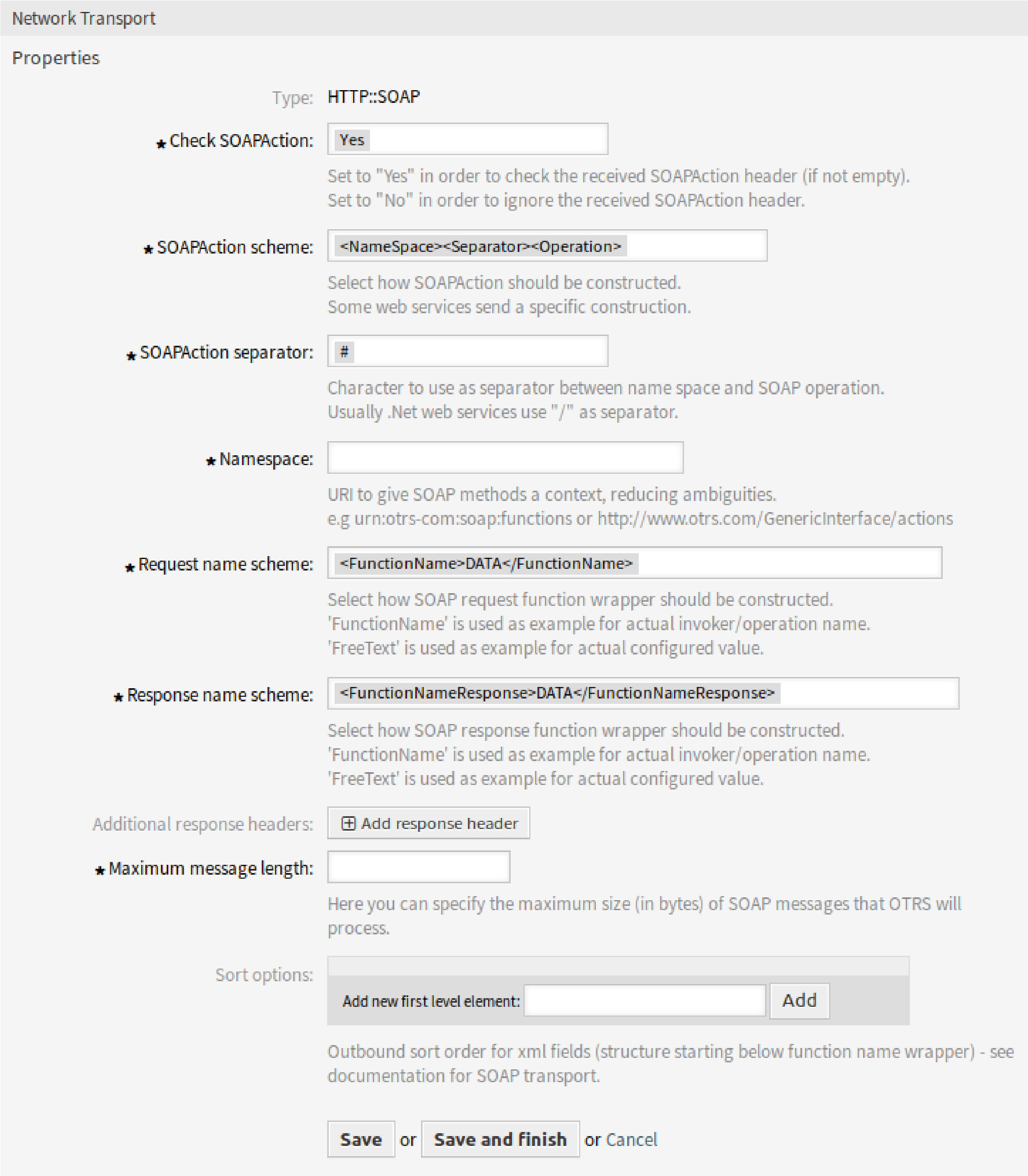
★ Use SSL Options: No

Show or hide SSL options to connect to the remote system.

Default command: GET

The default HTTP command to use for the requests.

Fig. 97: Web Service Settings - OTRS as Requester - HTTP::REST



Network Transport

Properties

Type: HTTP::SOAP

★ Check SOAPAction: Yes

Set to "Yes" in order to check the received SOAPAction header (if not empty).
Set to "No" in order to ignore the received SOAPAction header.

★ SOAPAction scheme: <NameSpace><Separator><Operation>

Select how SOAPAction should be constructed.
Some web services send a specific construction.

★ SOAPAction separator: #

Character to use as separator between name space and SOAP operation.
Usually .Net web services use "/" as separator.

★ Namespace:

URI to give SOAP methods a context, reducing ambiguities.
e.g urn:otrs-com:soap:functions or http://www.otrs.com/GenericInterface/actions

★ Request name scheme: <FunctionName>DATA</FunctionName>

Select how SOAP request function wrapper should be constructed.
'FunctionName' is used as example for actual invoker/operation name.
'FreeText' is used as example for actual configured value.

★ Response name scheme: <FunctionNameResponse>DATA</FunctionNameResponse>

Select how SOAP response function wrapper should be constructed.
'FunctionName' is used as example for actual invoker/operation name.
'FreeText' is used as example for actual configured value.

Additional response headers:

★ Maximum message length:

Here you can specify the maximum size (in bytes) of SOAP messages that OTRS will process.

Sort options:

Add new first level element:

Outbound sort order for xml fields (structure starting below function name wrapper) - see documentation for SOAP transport.

Fig. 98: Web Service Settings - OTRS as Requester - HTTP::SOAP

CHAPTER 6

Agent Interface

Completely new user interface has been developed for agents. This setting makes it possible to edit the agent styles that are selectable for all agents.

6.1 Styles

Use this screen to manage styles and choose style variants for use in agent interface. A fresh OTRS installation already contains some variants by default. The style management screen is available in the *Styles* module of the *Agent Interface* group.

The management screen consists of two widget. In the *Edit Styles* widget can be selected the styles for editing. The *Defaults* widget defines, which variant will be used as default for the agent interface. The agents can override this setting and they can select a different variant in they personal preferences, but they can not edit the styles.

6.1.1 Manage Styles

Styles are grouped into four categories: *Bright*, *High Contrast Bright*, *Dark* and *High Contrast Dark*.

To edit a style, select a category first. The edit screen will be opened for the selected category.

The *Edit Layout* section is the same for all categories.

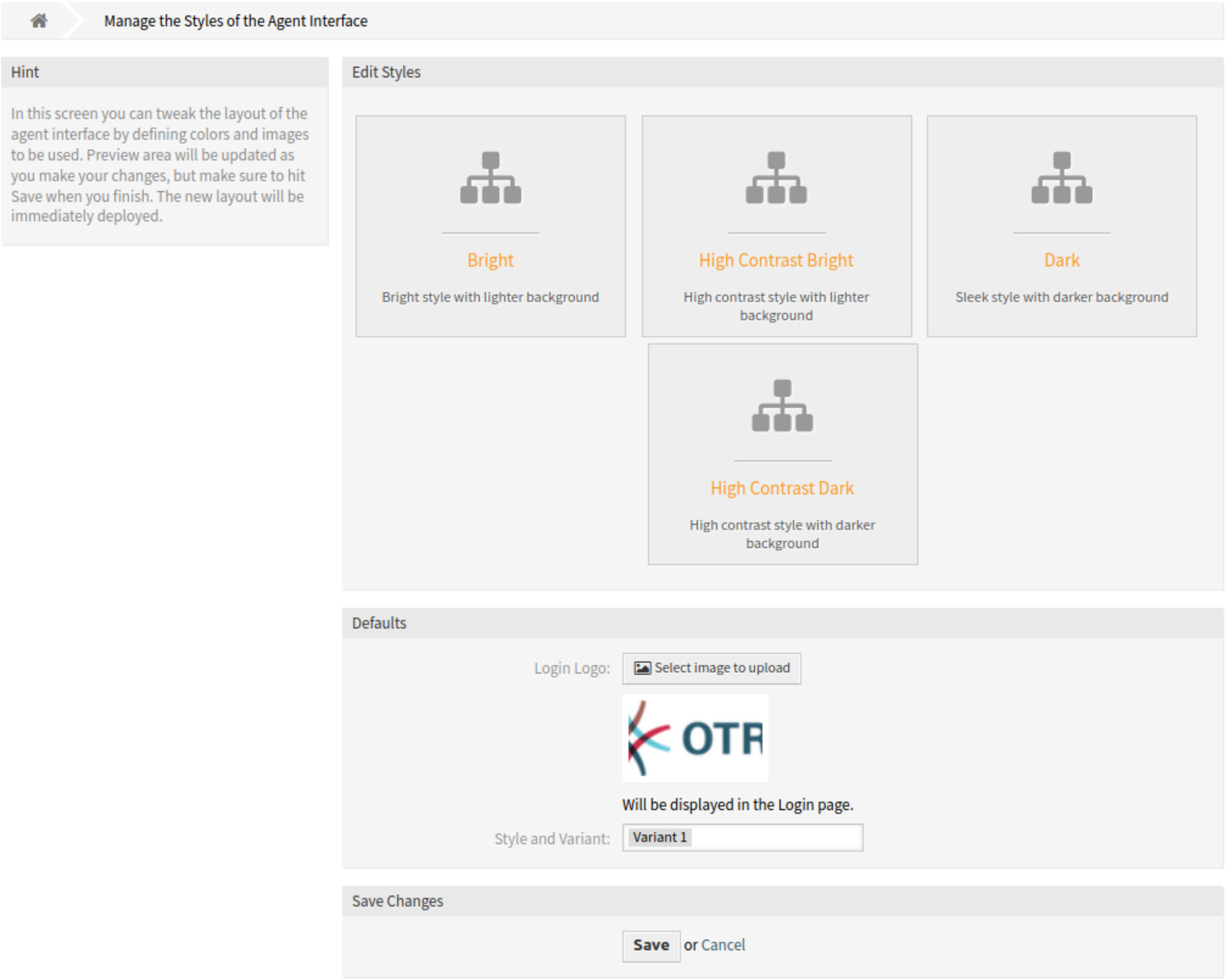
The following settings are available when adding or editing this resource.

Enable Select whether the style is available for agents.

Header Logo The logo is a small image that is displayed in the header of all pages.

> To change the logo, click on the *Select image to upload* button, and select a new logo image. Recommended file format is PNG.

Favicon The favorite icon is an icon that is displayed in the URL bar of the web browser.



Manage the Styles of the Agent Interface

Hint

In this screen you can tweak the layout of the agent interface by defining colors and images to be used. Preview area will be updated as you make your changes, but make sure to hit Save when you finish. The new layout will be immediately deployed.

Edit Styles

Bright

Bright style with lighter background

High Contrast Bright

High contrast style with lighter background

Dark

Sleek style with darker background

High Contrast Dark

High contrast style with darker background

Defaults

Login Logo: Select image to upload

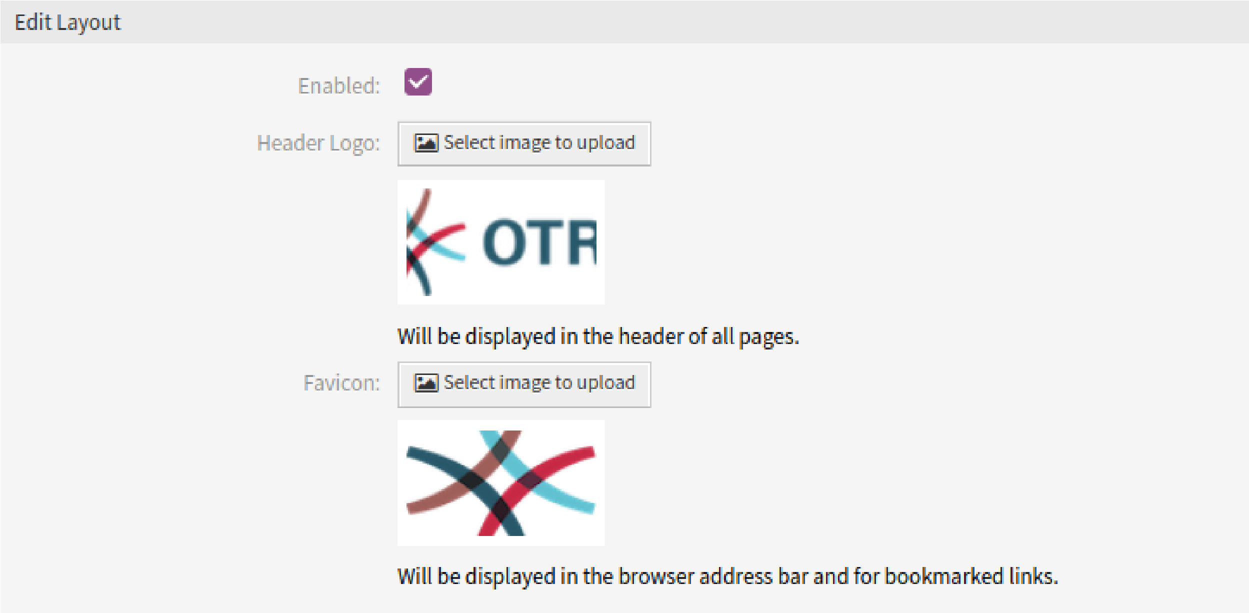
Will be displayed in the Login page.

Style and Variant: Variant 1

Save Changes

Save or Cancel

Fig. 1: Agent Style Management Screen



Edit Layout

Enabled: ☑

Header Logo: Select image to upload

Will be displayed in the header of all pages.

Favicon: Select image to upload

Will be displayed in the browser address bar and for bookmarked links.

Fig. 2: Agent Style Layout Edit Screen

To change the favorite icon, click on the *Select image to upload* button, and select a new icon. This is usually a 16×16 pixel image in PNG or ICO format.

Some categories contain different variants. Each variant can be customize in the same manner.

The following settings are available when adding or editing this resource.

Enable Select whether the variant is available for agents.

Name The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces.

Primary Color This color will be used as the background color of primary buttons and the color of textual links and hovered icons.

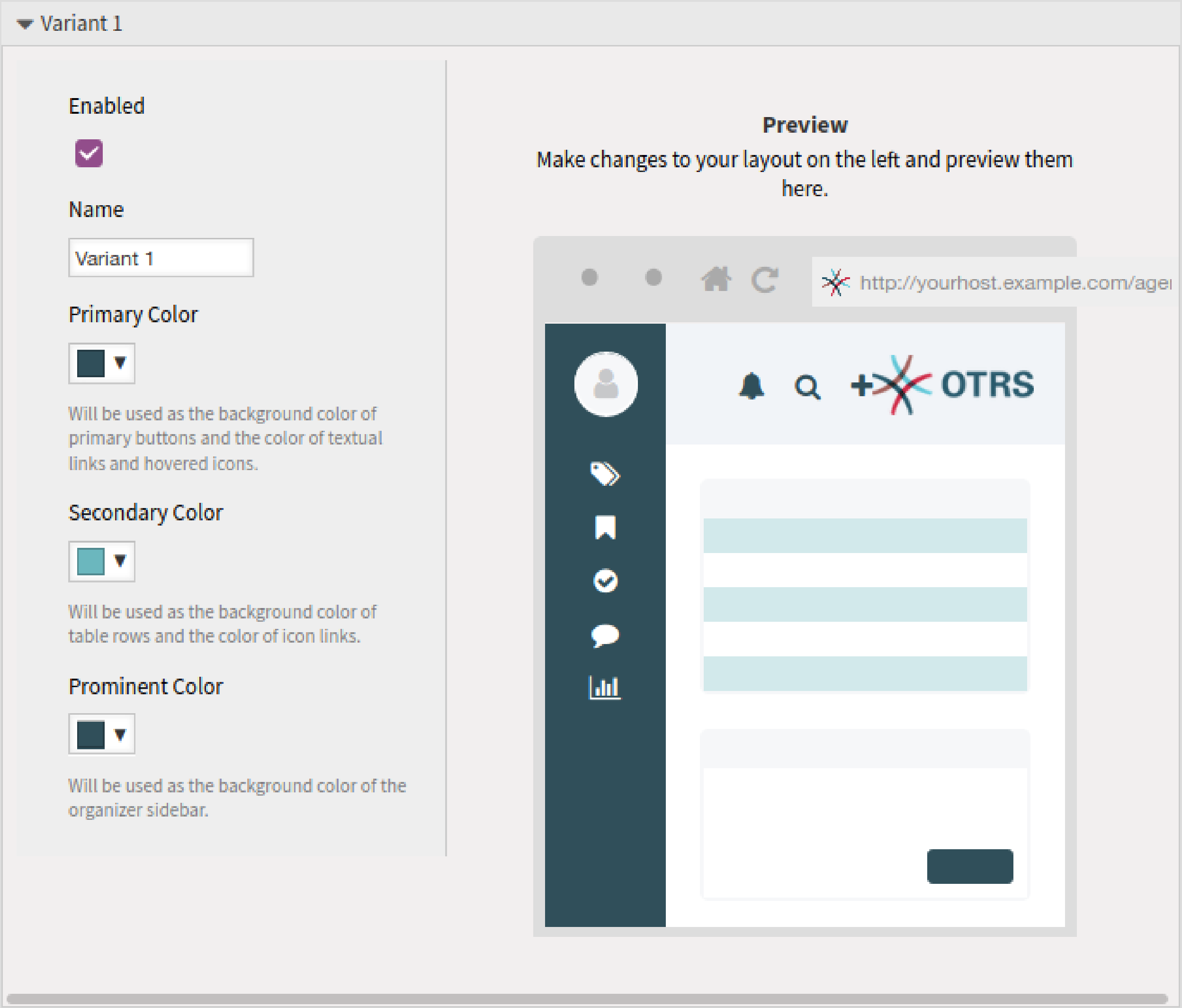
To change the primary color, just select a new color from the color palette. You can chose from the preselected colors or define other colors by choosing it from the color selector or typing the hexadecimal value. The new color will be displayed in the preview widget immediately.

Secondary Color This color will be used as the background color of table rows and the color of icon links.

To change the secondary color, just select a new color from the color palette. You can chose from the pre-selected colors or define other colors by choosing it from the color selector or typing the hexadecimal value. The new color will be displayed in the preview widget immediately.

Prominent Color This color will be used as the background color of the organizer sidebar.

To change the prominent color, just select a new color from the color palette. You can chose from the pre-selected colors or define other colors by choosing it from the color selector or typing the hexadecimal value. The new color will be displayed in the preview widget immediately.



▾ Variant 1

Enabled

☑

Name

Variant 1

Primary Color

■ ▾

Will be used as the background color of primary buttons and the color of textual links and hovered icons.

Secondary Color

■ ▾

Will be used as the background color of table rows and the color of icon links.

Prominent Color

■ ▾

Will be used as the background color of the organizer sidebar.

Preview

Make changes to your layout on the left and preview them here.

Fig. 3: Agent Style Variant

CHAPTER 7

External Interface

The need to transport valuable information about service to users is a major one for a service desk. Not always is it possible or plausible to do this via a corporate internet. The customers of a service desk need a one-stop solution for all their service needs.

OTRS provides access to create and manage demands, as well as receive valuable information such as the service catalog and knowledge base.

The following chapter describes the administration tools needed to implement corporate identity and manage language-based content.

7.1 Customer Service Catalogue

Use this screen to add categories and items for use in external interface. A fresh OTRS installation doesn't contain any categories or items by default. The catalogue management screen is available in the *Customer Service Catalogue* module of the *External Interface* group.

This module consists of two management screens: a category management screen and an item management screen.

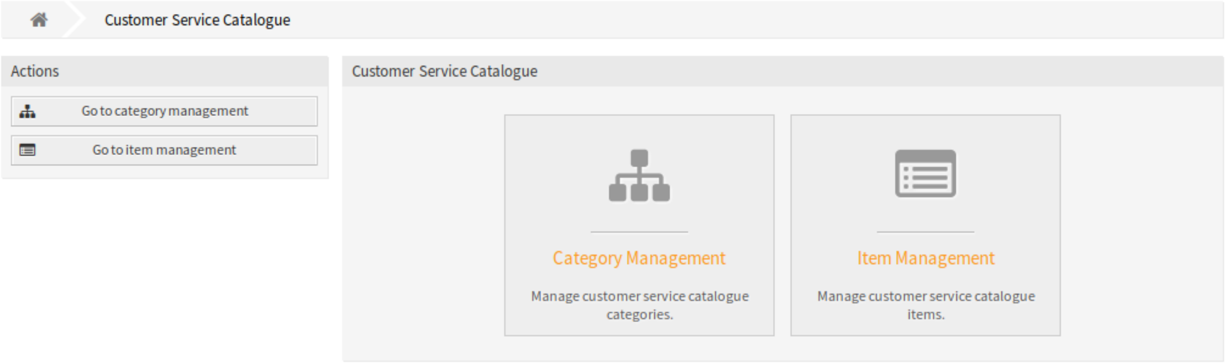
7.1.1 Manage Categories

Use this screen to add categories to collect the same items into groups. The *Category Management* screen in available via the *Go to category management* button or via the *Category Management* module.

To add a category:

1. Click on the *Add Category* button in the left sidebar.
2. Fill in the required fields.
3. Click on the *Save* button.

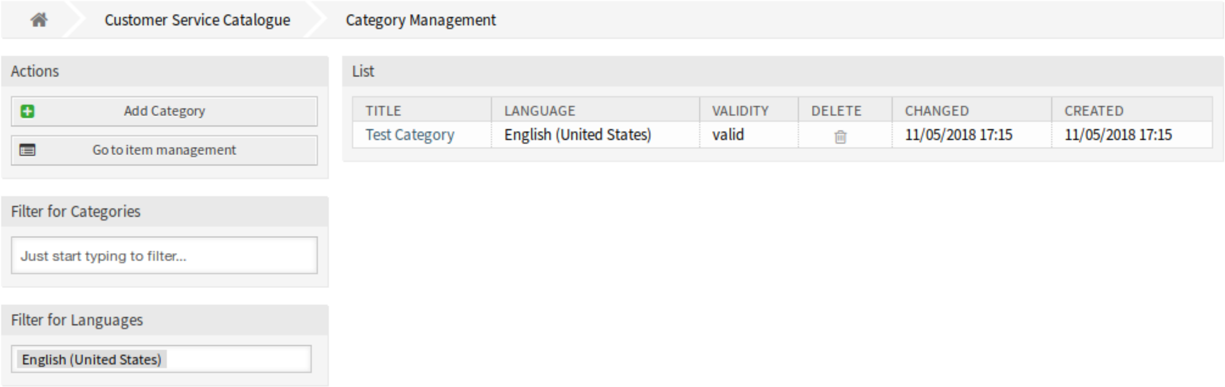
To edit a category:


Customer Service Catalogue
Actions
Go to category management
Go to item management
Customer Service Catalogue
Category Management
Item Management

Manage customer service catalogue categories.

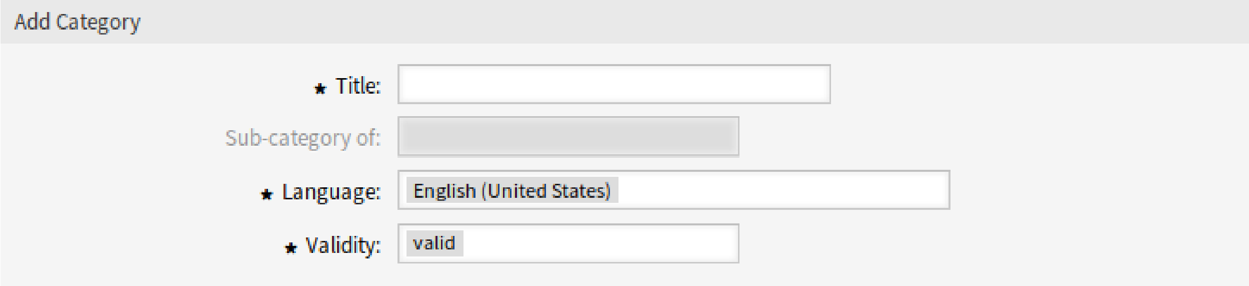
Manage customer service catalogue items.

Fig. 1: Customer Service Catalogue Management Screen



TITLE	LANGUAGE	VALIDITY	DELETE	CHANGED	CREATED
Test Category	English (United States)	valid	🗑	11/05/2018 17:15	11/05/2018 17:15

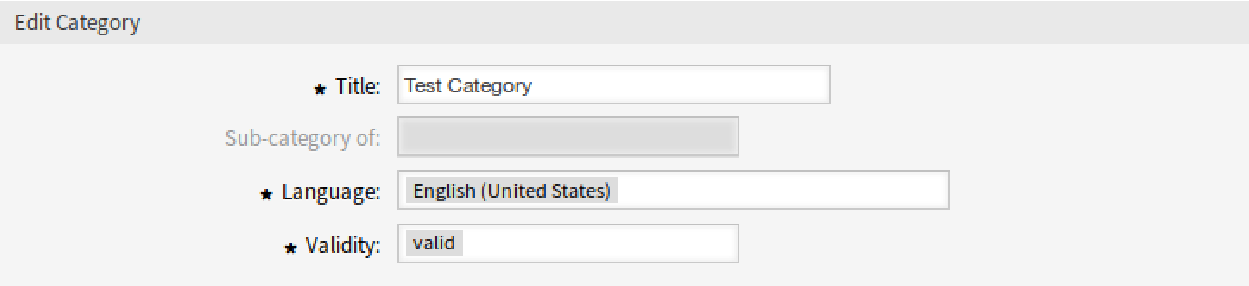
Fig. 2: Category Management Screen



★ Title:	
Sub-category of:	
★ Language:	English (United States)
★ Validity:	valid

Fig. 3: Add Category Screen

1. Click on a category in the list of categories.
2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.



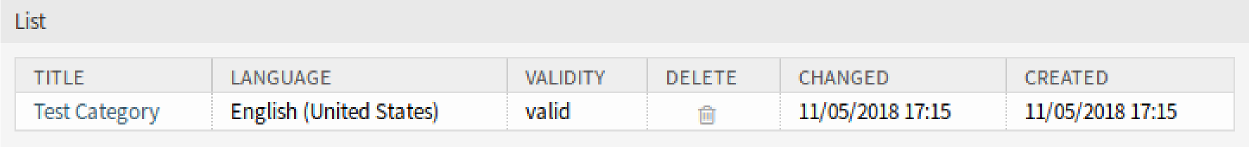
Edit Category

★ Title:	Test Category
Sub-category of:	
★ Language:	English (United States)
★ Validity:	valid

Fig. 4: Edit Category Screen

To delete a category:

1. Click on the trash icon in the *Delete* column of the overview table.
2. Click on the *Confirm* button.



List

TITLE	LANGUAGE	VALIDITY	DELETE	CHANGED	CREATED
Test Category	English (United States)	valid	🗑	11/05/2018 17:15	11/05/2018 17:15

Fig. 5: Delete Category Screen

Note: If several categories are added to the system, use the filter box to find a particular category by just typing the name to filter.

Category Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Title * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

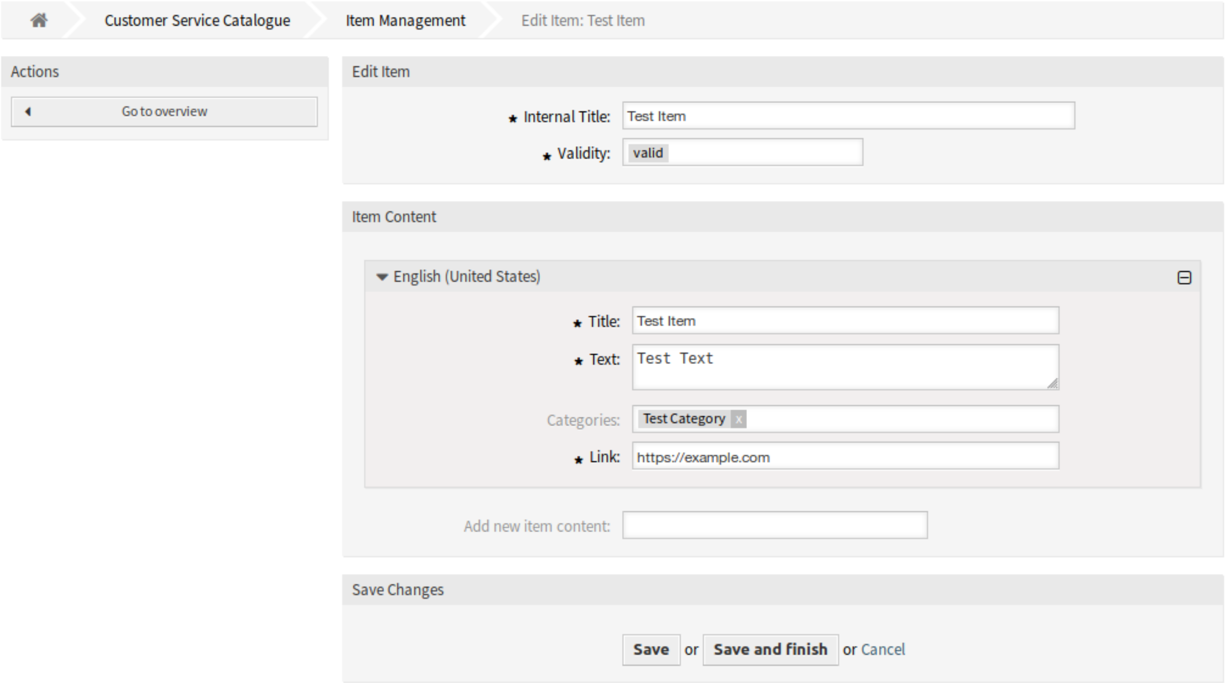
Sub-category of It is possible to add the new category under an existing one as sub-category. This will be displayed as *Parent Category::Child Category*.

Language * Select a language from the available languages of the system.

Validity * Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.

7.1.2 Manage Items

Use this screen to add items to the catalogue. Items can be collected into categories. The *Item Management* screen in available via the *Go to item management* button or via the *Item Management* module.



★ Internal Title: Test Item

★ Validity: valid

★ Title: Test Item

★ Text: Test Text

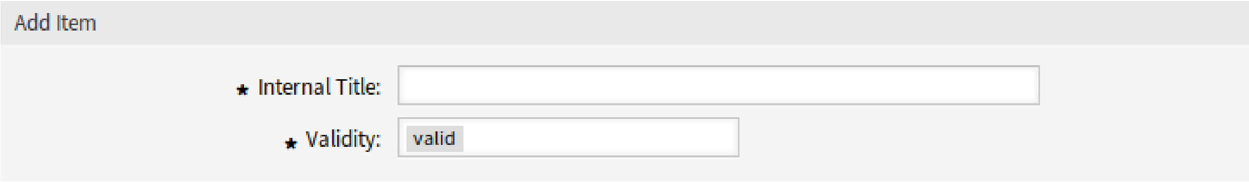
Categories: Test Category

★ Link: https://example.com

Fig. 6: Item Management Screen

To add an item:

1. Click on the *Add Item* button in the left sidebar.
2. Fill in the required fields.
3. Click on the *Save* button.



★ Internal Title:

★ Validity: valid

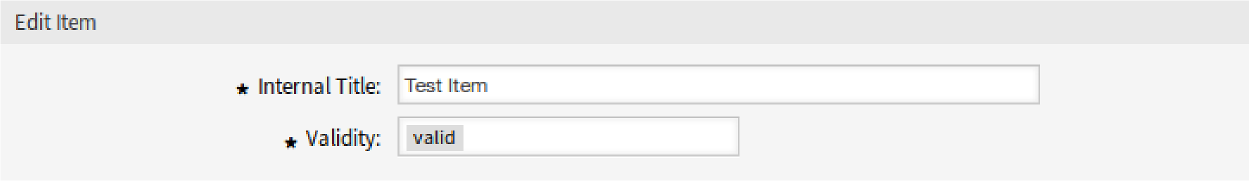
Fig. 7: Add Item Screen

To edit an item:

1. Click on an item in the list of items.
2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.

To delete an item:

1. Click on the trash icon in the *Delete* column of the overview table.
2. Click on the *Confirm* button.



Edit Item

⋆ Internal Title: Test Item

⋆ Validity: valid

Fig. 8: Edit Item Screen

Edit Item

⋆ Internal Title: Test Item

⋆ Validity: valid

Fig. 9: Delete Item Screen

Note: If several items are added to the system, use the filter box to find a particular item by just typing the name to filter.

Item Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Internal Title * The name of this resource, that is only displayed in the administrator interface. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Validity * Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.

Item Content In this widget can be added some localized content for the item.

Title * The name of this resource in the given language. Any type of characters can be entered to this field including uppercase letters and spaces.

Text * The text for this item in the given language.

Categories One ore more categories can be selected for the item in which the item should be visible.

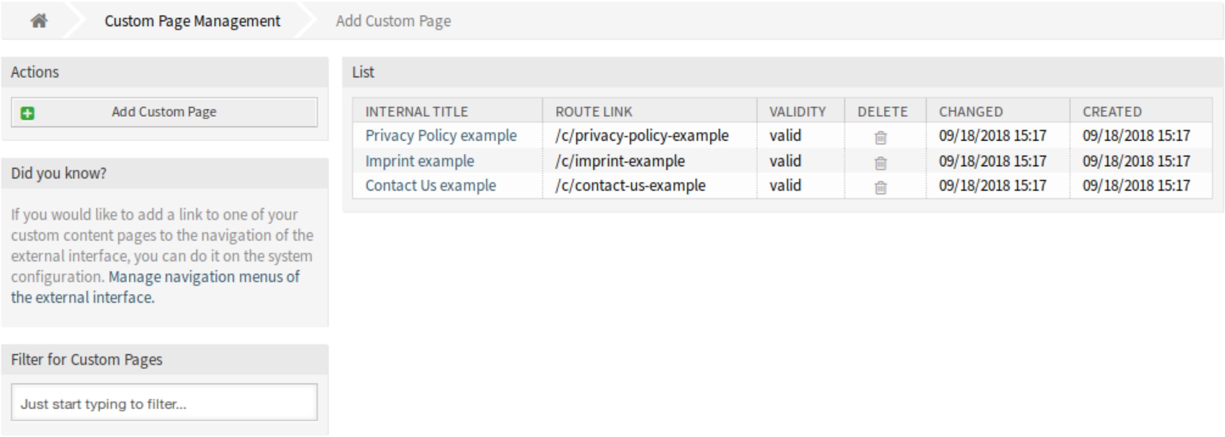
Note: Only those categories can be selected, that have the same language as the selected language for this widget.

Link * A link to an internal or an external URL.

Add new item content Select which languages should be added to create localized item content. All added languages can hold its own localized content, that are explained above.

7.2 Custom Pages

Use this screen to add custom pages for use in external interface. A fresh OTRS installation already contains some custom pages by default. The custom page management screen is available in the *Custom Pages* module of the *External Interface* group.



If you would like to add a link to one of your custom content pages to the navigation of the external interface, you can do it on the system configuration. Manage navigation menus of the external interface.

INTERNAL TITLE	ROUTE LINK	VALIDITY	DELETE	CHANGED	CREATED
Privacy Policy example	/c/privacy-policy-example	valid	🗑	09/18/2018 15:17	09/18/2018 15:17
Imprint example	/c/imprint-example	valid	🗑	09/18/2018 15:17	09/18/2018 15:17
Contact Us example	/c/contact-us-example	valid	🗑	09/18/2018 15:17	09/18/2018 15:17

Fig. 10: Custom Page Management Screen

7.2.1 Manage Custom Pages

Warning: Make sure to save your changes when you finish. The new configuration will be immediately deployed.

To add a custom page:

1. Click on the *Add Custom Page* button in the left sidebar.
2. Fill in the required fields.
3. Click on the *Save* button.

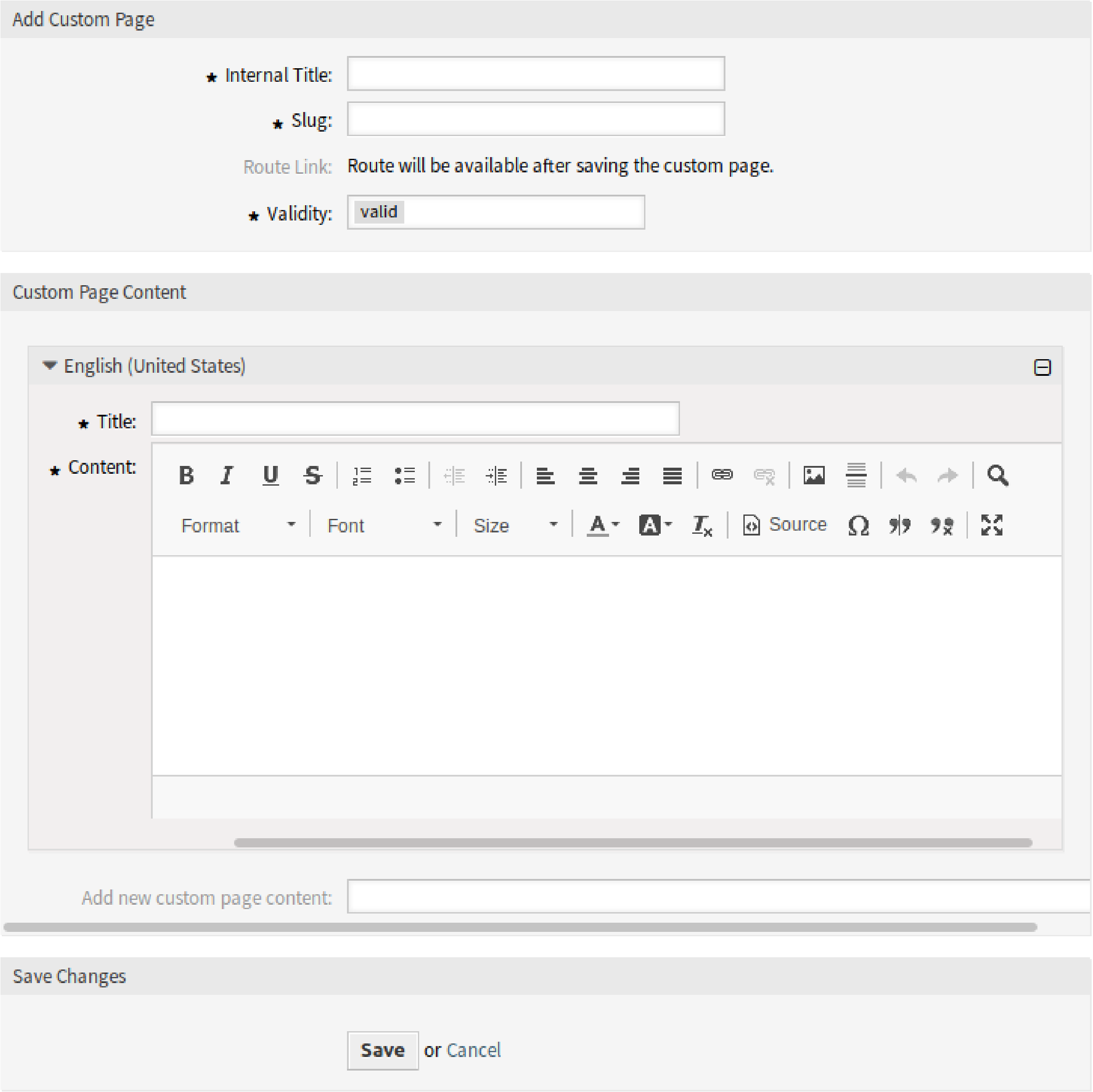
To edit a custom page:

1. Click on a custom page in the list of custom pages.
2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.

To delete a custom page:

1. Click on the trash icon in the fourth column of the overview table.
2. Click on the *Confirm* button.

Note: If several custom pages are added to the system, use the filter box to find a particular custom page by just typing the name to filter.



Add Custom Page

★ Internal Title:

★ Slug:

Route Link: Route will be available after saving the custom page.

★ Validity: valid

Custom Page Content

English (United States)

★ Title:

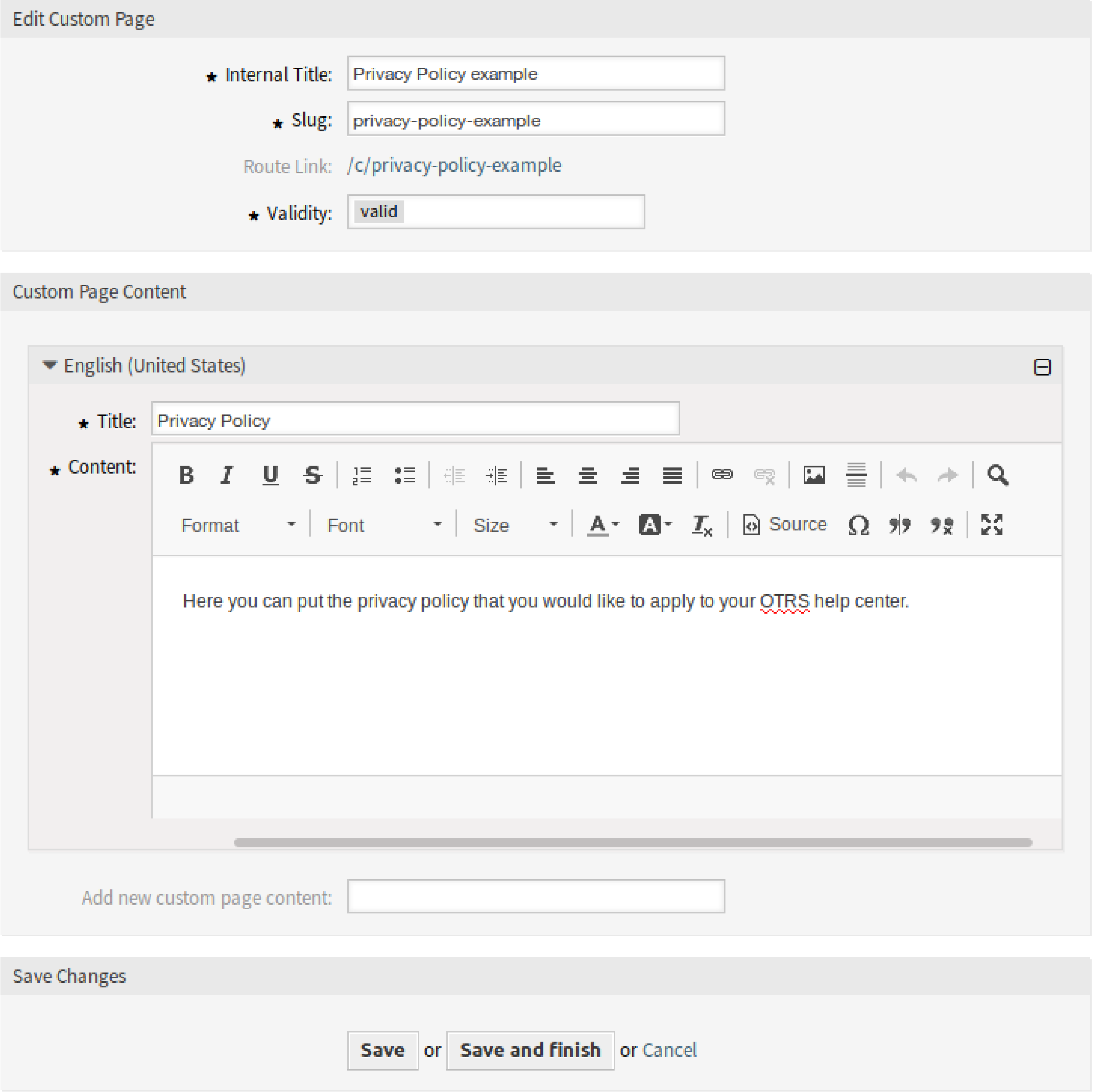
★ Content:

Add new custom page content:

Save Changes

Save or Cancel

Fig. 11: Add Custom Page Screen



Edit Custom Page

★ Internal Title:	Privacy Policy example
★ Slug:	privacy-policy-example
Route Link:	/c/privacy-policy-example
★ Validity:	valid

Custom Page Content

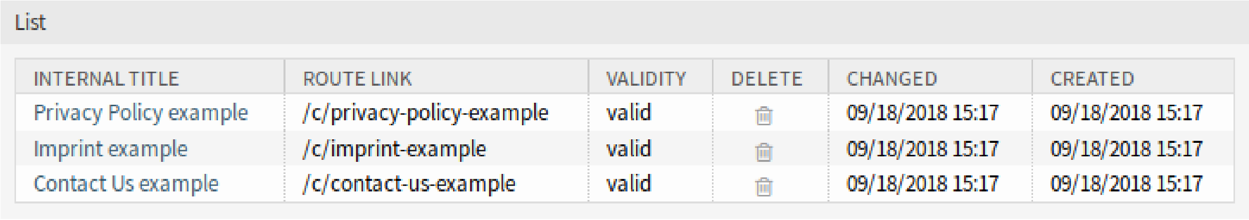
English (United States)

★ Title:	Privacy Policy
★ Content:	Here you can put the privacy policy that you would like to apply to your OTRS help center.

Add new custom page content:

Save Changes

Fig. 12: Edit Custom Page Screen



List

INTERNAL TITLE	ROUTE LINK	VALIDITY	DELETE	CHANGED	CREATED
Privacy Policy example	/c/privacy-policy-example	valid	🗑	09/18/2018 15:17	09/18/2018 15:17
Imprint example	/c/imprint-example	valid	🗑	09/18/2018 15:17	09/18/2018 15:17
Contact Us example	/c/contact-us-example	valid	🗑	09/18/2018 15:17	09/18/2018 15:17

Fig. 13: Delete Custom Page Screen

7.2.2 Custom Page Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Internal Title * The name of this resource, that is only displayed in the administrator interface. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Slug * This will be the URL of the custom page. Recommended characters are lowercase letters, numbers and minus sign.

Validity * Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.

Custom Page Content In this widget can be added some localized content for the item.

> **Title *** The name of this resource in the given language. Any type of characters can be entered to this field including uppercase letters and spaces.
>
> **Content *** The text for this item in the given language.
>
> **Add new custom page content** Select which languages should be added to create localized item content. All added languages can hold its own localized content, that are explained above.

7.3 Home Page

Use this screen to define home page configuration for different user languages, that are displayed in the external interface. The home page management screen is available in the *Home Page* module of the *External Interface* group.

This screen contains several widget for each languages, where localized content can be added.

> **Warning:** Make sure to save your changes when you finish. The new configuration will be immediately deployed.

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

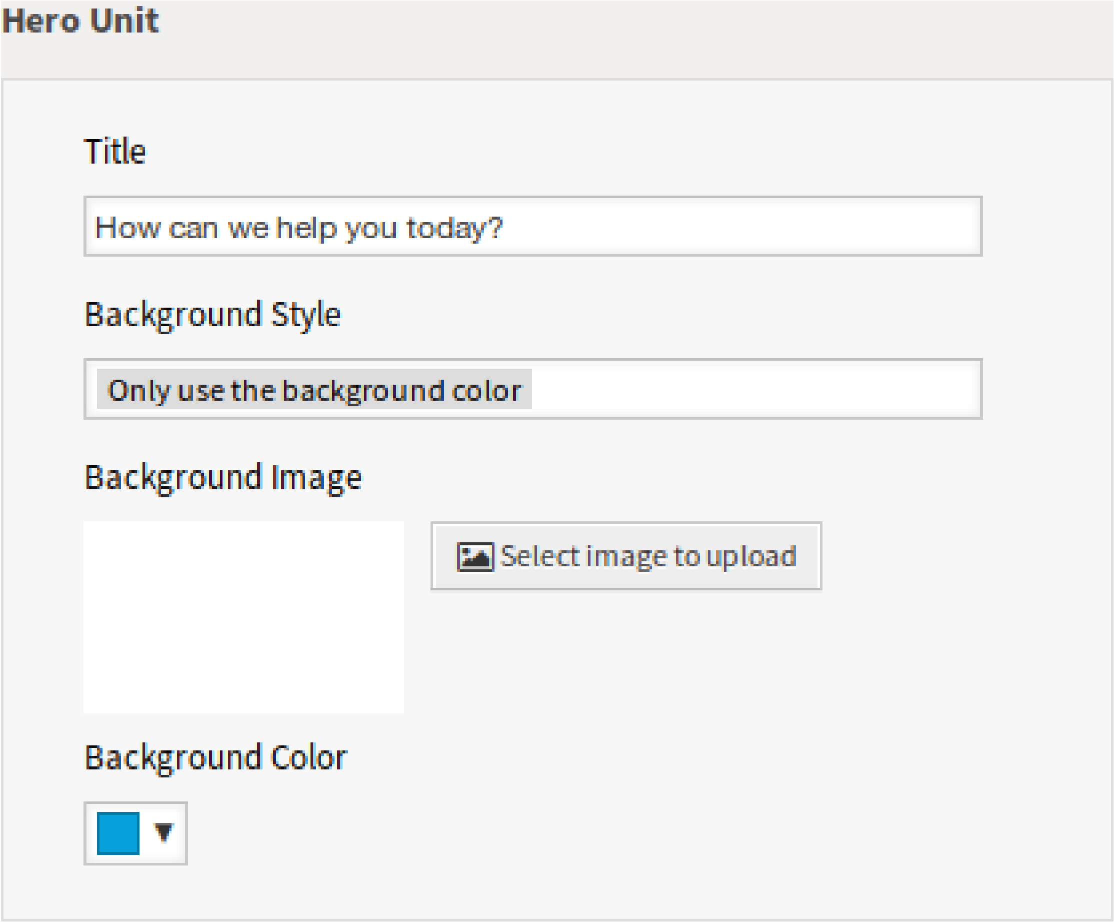
7.3.1 Hero Unit

Title This is the most important sentence or motto of the home page.

Background Style Define the behavior of background image and background color. The following options are available:

- Only use the background image
- Only use the background color
- Use the background image and overlay it with the selected color

Background Image Select an image used as background image for the hero unit. To select an image, click on the *Select image to upload* button and chose an image from your file system.



Hero Unit

Title

How can we help you today?

Background Style

Only use the background color

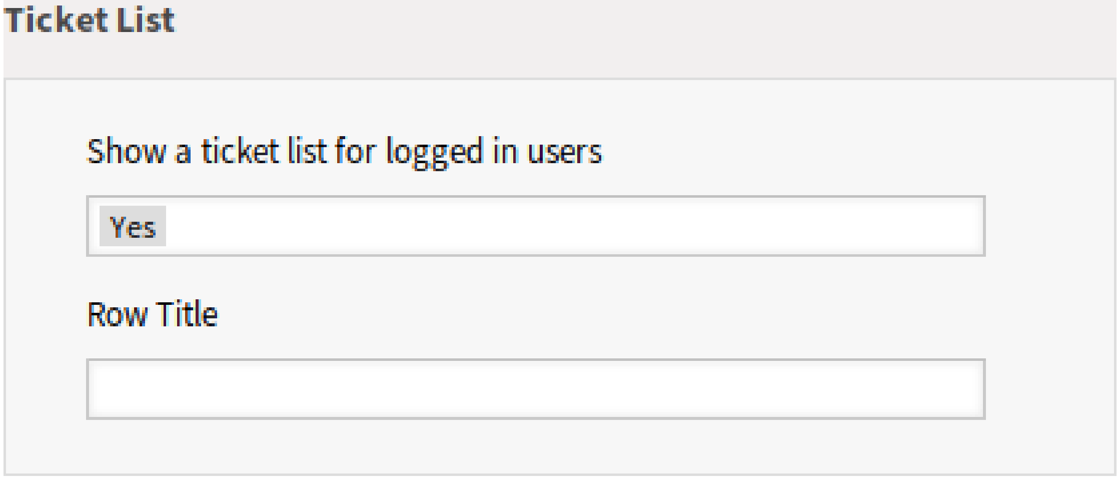
Background Image

Background Color

Fig. 14: Hero Unit Widget

Background Color To change the background color, just select a new color from the color palette. You can chose from the pre-selected colors or define other colors by choosing it from the color selector or typing the hexadecimal value.

7.3.2 Ticket List



Ticket List

Show a ticket list for logged in users

Yes

Row Title

Fig. 15: Ticket List Widget

Show a ticket list for logged in users This setting defines if the ticket list should be visible for logged in users or not.

Row Title This is an optional title for the row, that contains this widget in the external interface.

7.3.3 Image Teasers

The image teasers are displayed in a separate row in the external interface.

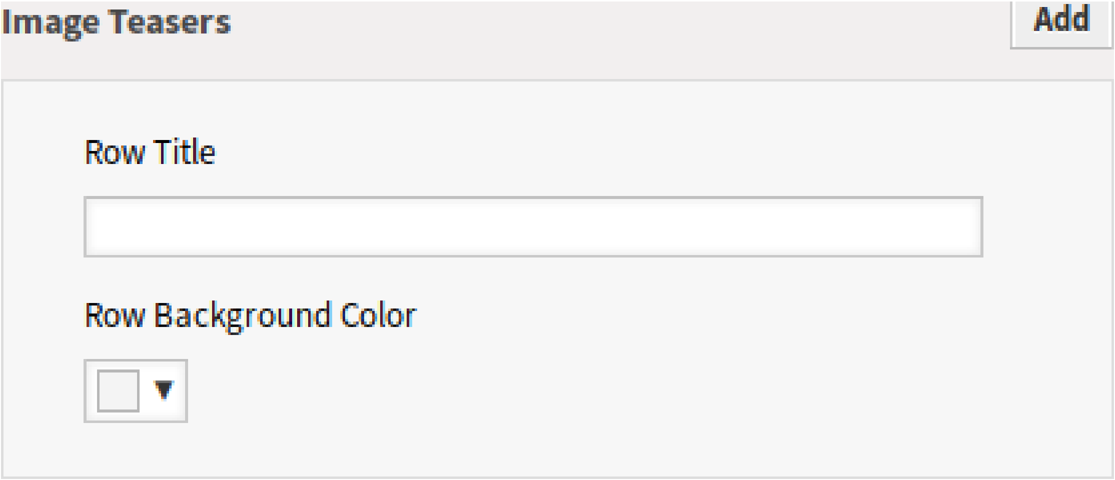


Image Teasers

Row Title

Row Background Color

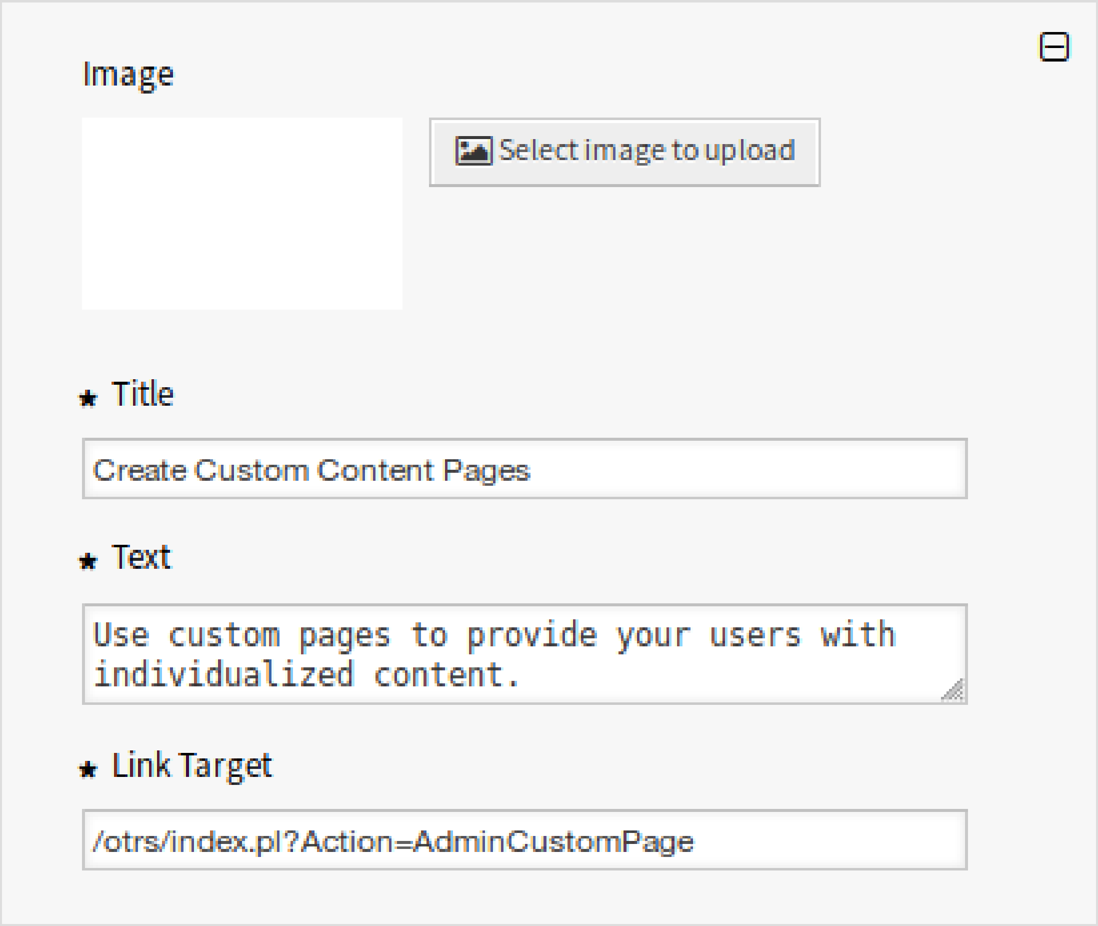
□ ▼

Fig. 16: Image Teasers Widget

Row Title This is an optional title for the row, that contains this widget in the external interface.

Row Background Color To change the background color, just select a new color from the color palette. You can chose from the pre-selected colors or define other colors by choosing it from the color selector or typing the hexadecimal value.

You can add up to three image teasers by clicking on the *Add* button in the top right corner of the widget.



Image

Select image to upload

★ Title

Create Custom Content Pages

★ Text

Use custom pages to provide your users with individualized content.

★ Link Target

/otrs/index.pl?Action=AdminCustomPage

Fig. 17: Image Teaser

Image Select an image to display it in the image teaser widget. To select an image, click on the *Select image to upload* button and chose an image from your file system.

Title * The heading text that are displayed in this item.

Text * The text for this item.

Link Target * An URL that will be opened after clicking on this item in the external interface.

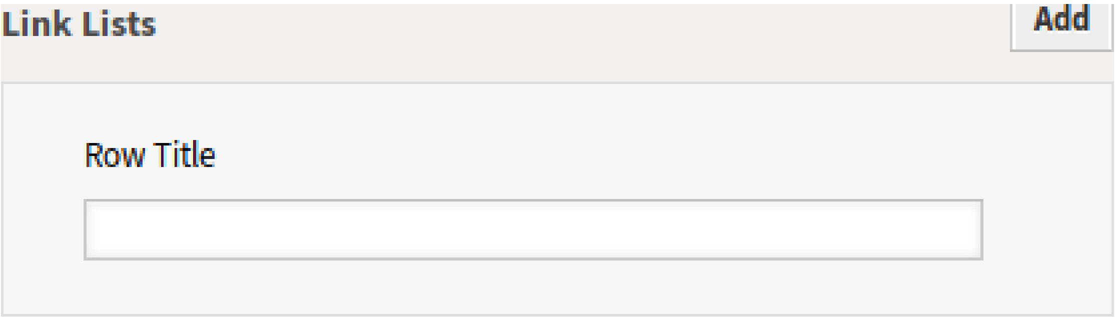
7.3.4 Link Lists

The link lists are displayed in a separate row in the external interface.

Row Title This is an optional title for the row, that contains this widget in the external interface.

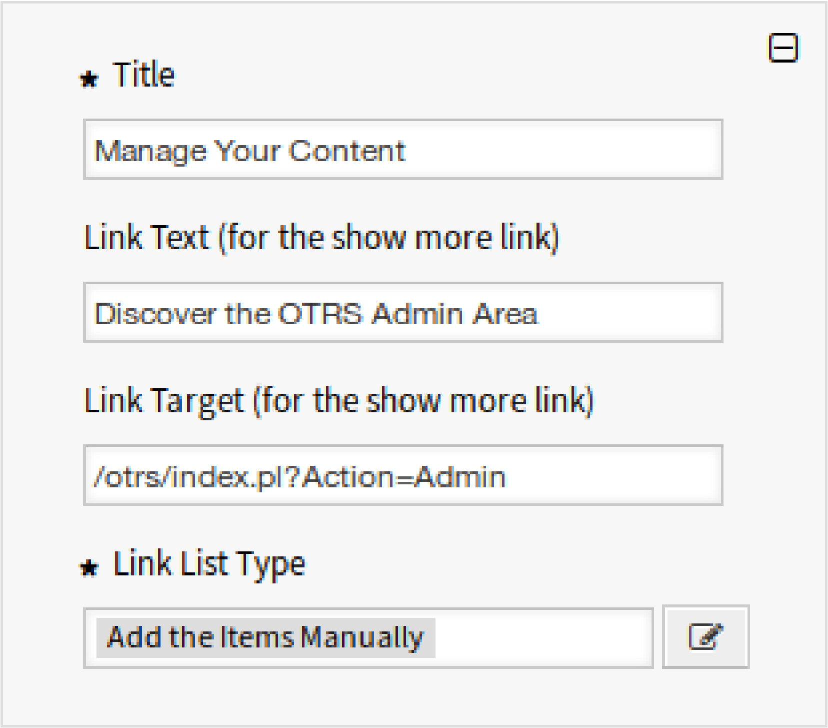
You can add up to four link lists by clicking on the *Add* button in the top right corner of the widget.

Title * The heading text that are displayed in this item.



Row Title

Fig. 18: Link Lists Widget



★ Title

Manage Your Content

Link Text (for the show more link)

Discover the OTRS Admin Area

Link Target (for the show more link)

/otrs/index.pl?Action=Admin

★ Link List Type

Add the Items Manually

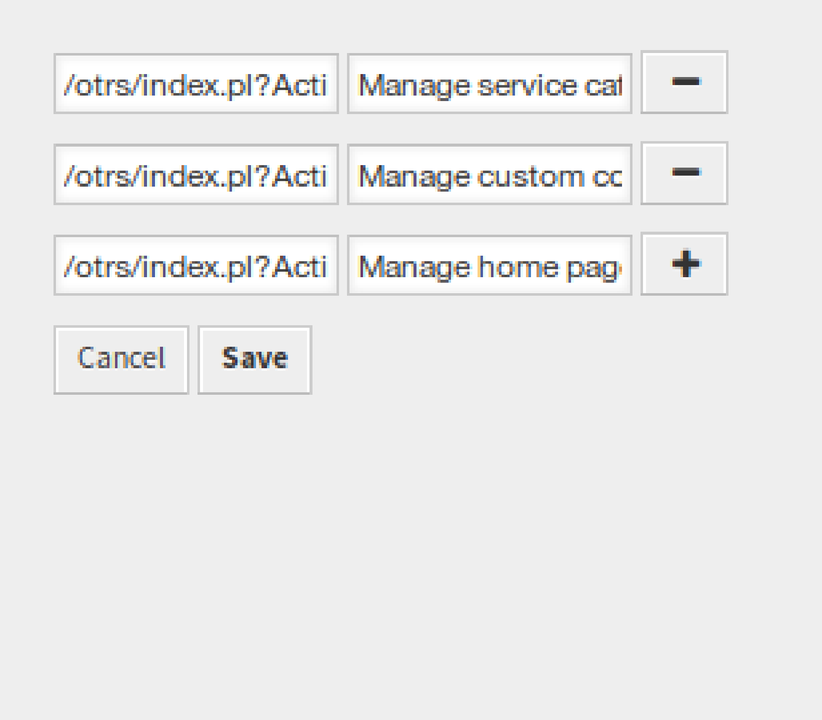
Fig. 19: Link List

Link Text Add the text for the *show more* link.

Link Target Add the URL for the *show more* link.

Link List Type * Specifies the functionality how the item will be added.

Add the Items Manually Clicking on the edit button a new widget will be visible to add the items.



/otrs/index.pl?Acti | Manage service cat | −

/otrs/index.pl?Acti | Manage custom cc | −

/otrs/index.pl?Acti | Manage home pag | +

Fig. 20: Add the Items Manually

Show the Results of a Search Clicking on the edit button a new widget will be visible to add the items.

7.3.5 Content Cards

The link lists are displayed in a separate row in the external interface.

Row Title This is an optional title for the row, that contains this widget in the external interface.

You can add up to three content cards by clicking on the *Add* button in the top right corner of the widget.

Title * The heading text that are displayed in this item.

Text * The text for this item.

Link Text Add the text for the *show more* link.

Link Target An URL that will be opened after clicking on this item in the external interface.

7.4 Layout

Use this screen to edit the layout displayed in external interface. The layout management screen is available in the *Layout* module of the *External Interface* group.

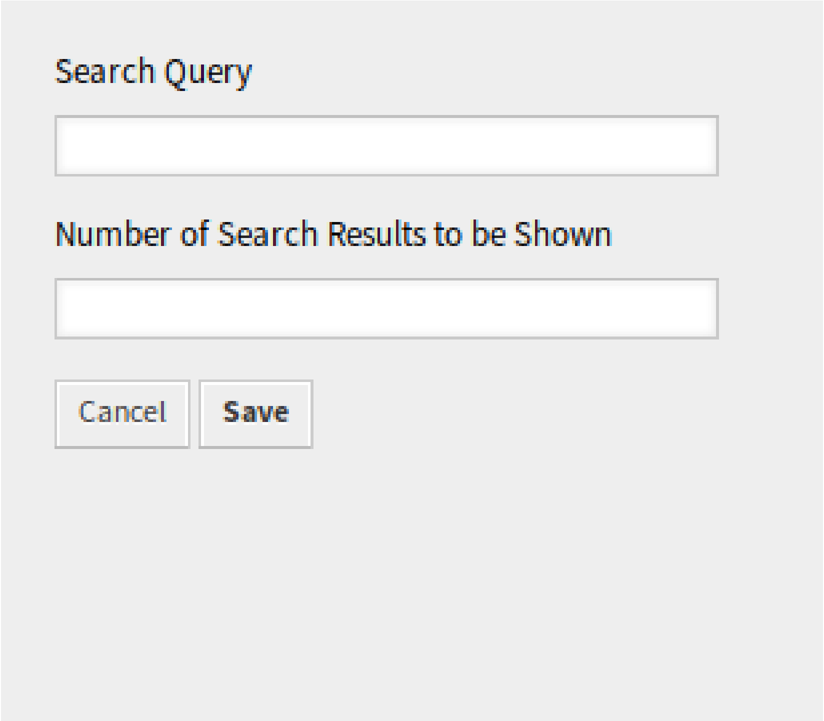


Fig. 21: Show the Results of a Search

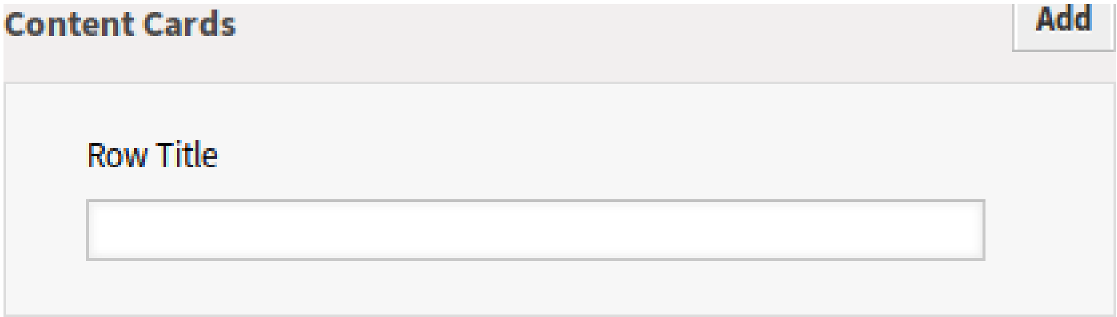
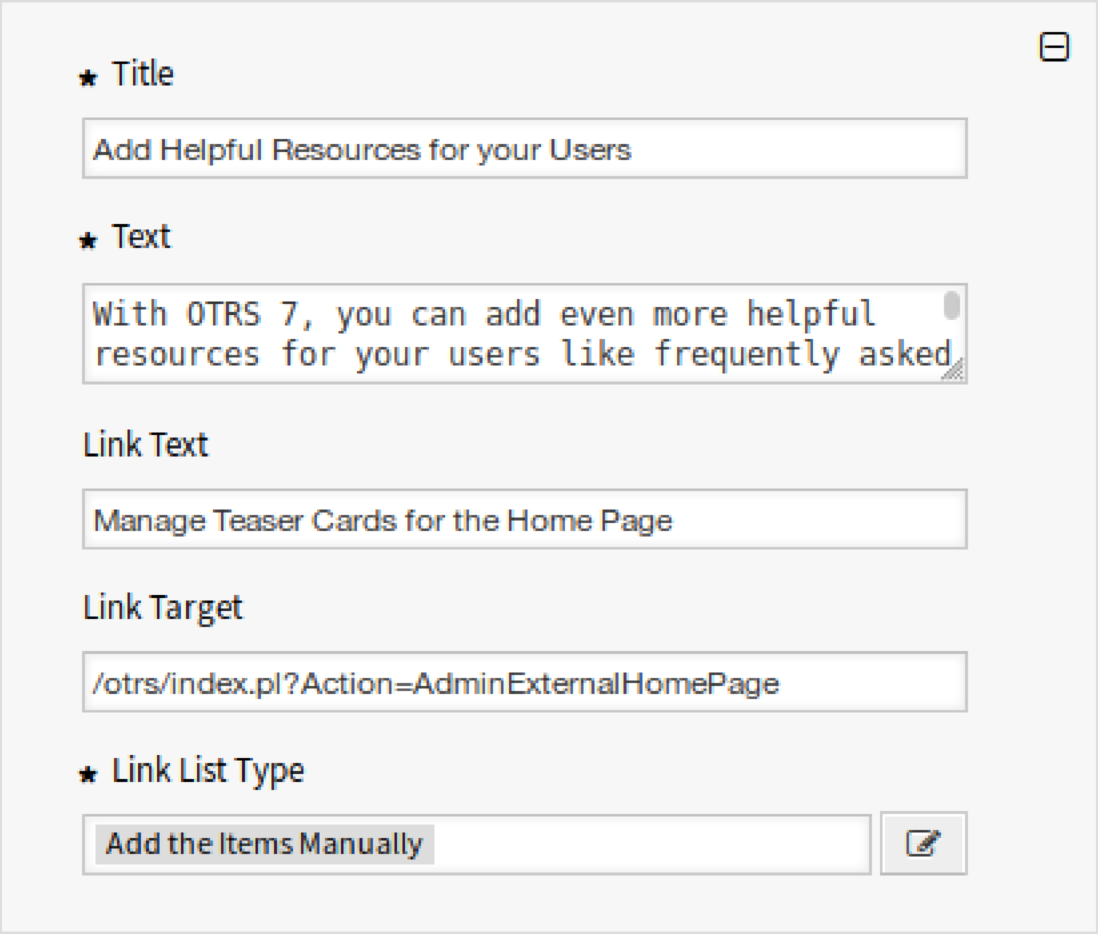


Fig. 22: Content Cards Widget



★ Title

Add Helpful Resources for your Users

★ Text

With OTRS 7, you can add even more helpful resources for your users like frequently asked

Link Text

Manage Teaser Cards for the Home Page

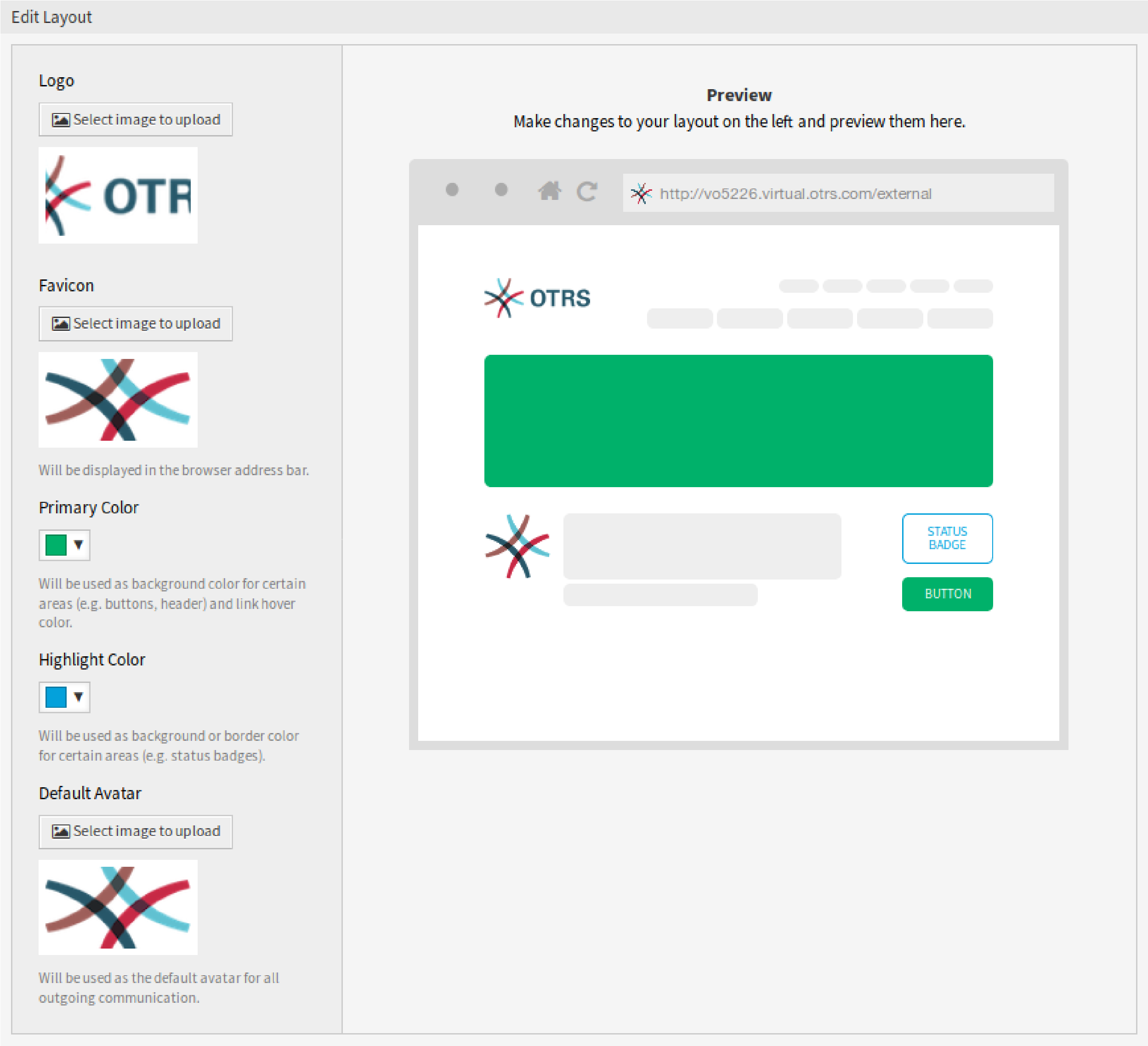
Link Target

/otrs/index.pl?Action=AdminExternalHomePage

★ Link List Type

Add the Items Manually

Fig. 23: Content Card



Edit Layout

Logo

Select image to upload

Favicon

Select image to upload

Will be displayed in the browser address bar.

Primary Color

🟩 ▼

Will be used as background color for certain areas (e.g. buttons, header) and link hover color.

Highlight Color

🟦 ▼

Will be used as background or border color for certain areas (e.g. status badges).

Default Avatar

Select image to upload

Will be used as the default avatar for all outgoing communication.

Preview

Make changes to your layout on the left and preview them here.

http://vo5226.virtual.otrs.com/external

Fig. 24: Edit Layout Screen

A fresh OTRS installation already contains a default layout. In this screen all parameters can be changed.

Warning: Make sure to save your changes when you finish. The new configuration will be immediately deployed.

Logo The logo is a small image that is displayed in the top left corner of the external pages.

To change the logo, click on the *Select image to upload* button, and select a new logo image. Recommended file format is PNG.

Favicon The favorite icon is an icon that is displayed in the URL bar of the web browser.

To change the favorite icon, click on the *Select image to upload* button, and select a new icon. This is usually a 16×16 pixel image in PNG or ICO format.

Primary Color Primary color is the most important color of the external pages (see the preview screen).

To change the primary color, just select a new color from the color palette. You can chose from the preselected colors or define other colors by choosing it from the color selector or typing the hexadecimal value. The new color will be displayed in the preview widget immediately.

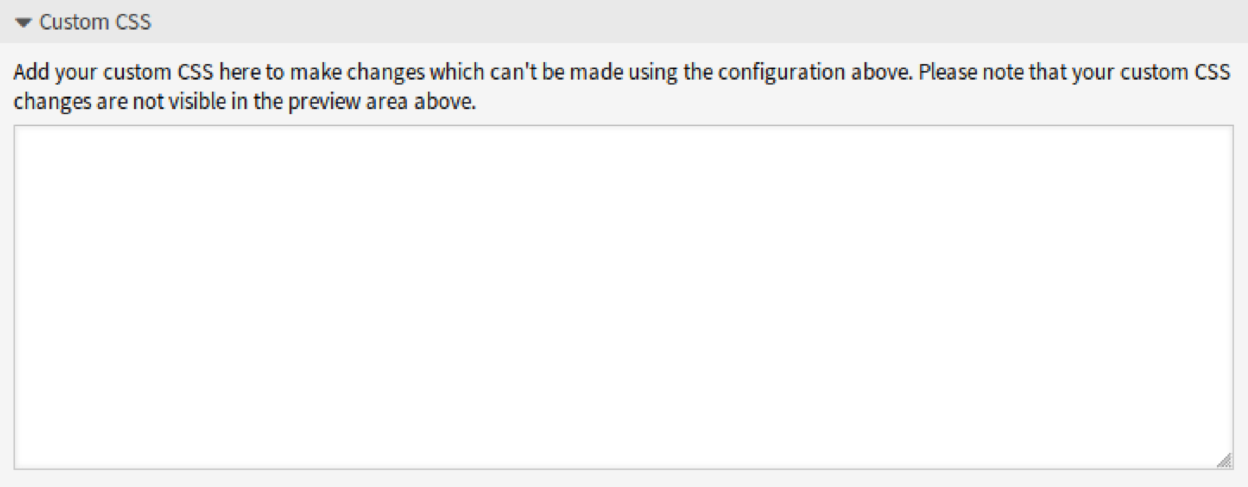
Highlight Color Highlight color is the second color of the external pages using for status badges, links, etc.

To change the highlight color, just select a new color from the color palette. You can chose from the preselected colors or define other colors by choosing it from the color selector or typing the hexadecimal value. The new color will be displayed in the preview widget immediately.

Default Avatar Will be used as the default avatar for all outgoing communication.

To change the avatar, click on the *Select image to upload* button, and select a new avatar image.

Custom CSS Use this text area to add custom CSS to be applied in the external interface.



Add your custom CSS here to make changes which can't be made using the configuration above. Please note that your custom CSS changes are not visible in the preview area above.

Fig. 25: Custom CSS Widget

7.4.1 Include Custom Scripts

External interface application has been built with security-first mindset in place. It contains several mechanisms to make sure that all code in the front end is provided directly by the OTRS framework.

In addition, a security standard called Content Security Policy headers is now leveraged by the built-in web server. It serves as a last line of defense and in case some unexpected code still makes it to the client, it will make sure that the user’ s browser simply refuses to run it.

However, a use-case still exists in which even a modern web application might still need to run some additional code, provided it is vetted and included by the administrators. One example could be an external integrated chat solution, but could also be an inline code snippet used for external web analytics.

See also:

Including custom scripts can be done in the *System Configuration* with the following settings:

- `ExternalFrontend::ExternalScripts`
- `ExternalFrontend::InlineScripts`
- `WebApp::Server::AdditionalOrigins`

Web Analytics Example

In this example, we outline the steps to include custom scripts used by a web analytics provider in the external interface application. They will be run each time the application is used in order to track and report the behavior of the user for later analysis.

We start with the provided code snippet below, from a third party provider.

```
<script async src="https://www.example.com/track/js?id=UA-XXXXXX-Y"></script>
<script>
   window.dataLayer = window.dataLayer || [];
   function track(){dataLayer.push(arguments);}
   track('js', new Date());

   track('config', 'UA-XXXXXX-Y');
</script>
```

First line of the snippet refers to an *external* resource which should be loaded by the `script` tag.

In the script block below, we have some *inline* code which is run immediately when the page is loaded.

Adding External Script Resources

First, we need to include the external script location in the appropriate configuration:

1. Go to the *System Configuration* screen.
2. Search for the setting `ExternalFrontend::ExternalScripts`.
3. Click on the setting in order to edit it.
4. In case a value is already present, click on the plus button. Otherwise, proceed below.
5. Enter the full location of the *external* resource in the text field. For example: `https://www.example.com/track/js?id=UA-XXXXXX-Y`.
6. Click on the check mark in order to save the modified setting.
7. Deploy the modified system configuration.

Adding Inline Code Snippets

Next, we have to also include the inline part of the original snippet:

1. Go to the *System Configuration* screen.
2. Search for the setting `ExternalFrontend::InlineScripts`.
3. Click on the setting in order to edit it.
4. In case a value is already present, click on the plus button. Otherwise, proceed below.
5. Enter the full code snippet in the text area, minus any `script` tags. For example:

```
window.dataLayer = window.dataLayer || [];
function track(){dataLayer.push(arguments);}
track('js', new Date());

track('config', 'UA-XXXXXX-Y');
```

6. Click on the check mark in order to save the modified setting.
7. Deploy the modified system configuration.

Rebuilding the External Interface Application

In order to apply the changes, we also need to rebuild the external interface application. Drop down to shell, and execute the following command:

```
bin/otrs.WebServer.pl --deploy-assets
```

Whitelisting Additional Origins in the Security Headers

If you now try to access the external interface application, you will be able to verify that the scripts are included in the code. However, your browser will probably block access to all *inline* and *external* resources, therefore the code might fail with some errors.

This behavior is by design, since external resources can only be loaded if they are specifically whitelisted in the *Content Security Policy* headers.

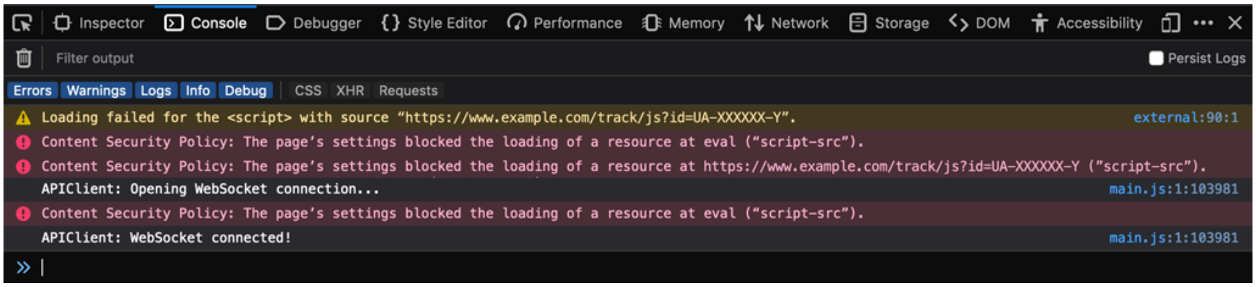
To check for blocked code, please use suitable web browser inspection tools. In our example, we will be using Mozilla Firefox and its web console available via *Tools* → *Web Developer* → *Web Console* menu item, or via the *F12* shortcut key.

For the example code snippet, you might receive following errors in the console when the application is accessed:

From the console errors we can see that the external script resource was prevented from being loaded (lines 1 and 3). In addition to that, two evaluation calls were also blocked (lines 2 and 5). All errors reference a *Content Security Policy* rule under the name of `script-src`, which signals script resources.

We need to add both the external resource and the evaluation calls to the additional origins list of the *Content Security Policy* headers:

1. Go to the *System Configuration* screen.
2. Search for the setting `WebApp::Server::AdditionalOrigins`.
3. Click on the setting in order to edit it.



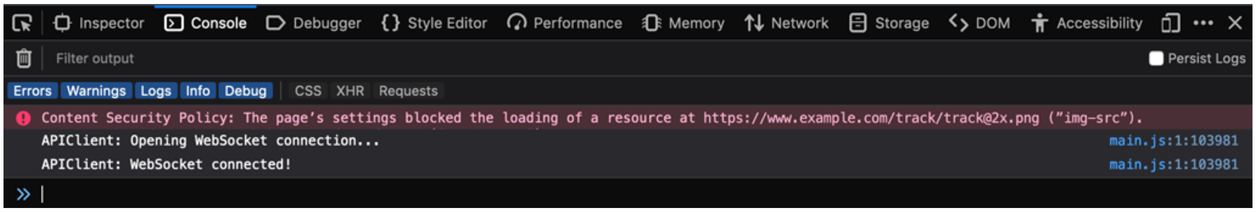
```
⚠ Loading failed for the <script> with source “https://www.example.com/track/js?id=UA-XXXXXX-Y”.                   external:90:1
❗ Content Security Policy: The page’s settings blocked the loading of a resource at eval (“script-src”).
❗ Content Security Policy: The page’s settings blocked the loading of a resource at https://www.example.com/track/js?id=UA-XXXXXX-Y (”script-src”).
  APIClient: Opening WebSocket connection...                                                                   main.js:1:103981
❗ Content Security Policy: The page’s settings blocked the loading of a resource at eval (“script-src”).
  APIClient: WebSocket connected!                                                                              main.js:1:103981
```

Fig. 26: Browser Console Errors and Warnings

4. In case a value for `script-src` is already present, click on the plus button next to it. Otherwise, proceed below.
5. Enter the domain part only of the blocked resource in the text field. For example: `https://www.example.com`. This allows the external resource to be loaded.
6. Click on the plus button next to the field, so another value is added.
7. Enter the following directive in the new field, including the quotes: `'unsafe-eval'`. This allows the evaluation calls to be executed.
8. Click on the check mark in order to save the modified setting.
9. Deploy the modified system configuration.

There is no need to rebuild the external interface application at this point, as the additional origins configuration should be immediately in effect.

If you reload the external interface application, you might get some additional errors. In our example, it might be the following:



```
❗ Content Security Policy: The page’s settings blocked the loading of a resource at https://www.example.com/track/track@2x.png (”img-src”).
  APIClient: Opening WebSocket connection...                                                                   main.js:1:103981
  APIClient: WebSocket connected!                                                                              main.js:1:103981
```

Fig. 27: Additional Browser Console Errors and Warnings

This error points that an additional resource that was also blocked, an image at a specific location (line 1). We can deduce this via the name of the *Content Security Policy* rule `img-src`, which references an image resource. In order to add it to the whitelist, try the following:

1. Go to the *System Configuration* screen.
2. Search for the setting `WebApp::Server::AdditionalOrigins`.
3. Click on the setting in order to edit it.
4. In case a value for `img-src` is already present, click on the plus button next to it. Otherwise, proceed below.
5. Enter the domain part only of the blocked resource in the text field. For example: `https://www.example.com`. This allows the external image resource to be loaded.
6. Click on the check mark in order to save the modified setting.

7. Deploy the modified system configuration.

Try again to reload the external interface application and check if there are more errors. If not, your scripts are now probably working as expected.

Unfortunately, it is not possible to predict what kind of resources your scripts might be requiring. But, no worries, you can whitelist most of them, just make sure to follow the trail of hints shown in the browser console log. Find a corresponding header rule in the configuration and update it accordingly.

Note: Some resources might only be requested by others, hence several iterations might be needed until everything is configured properly.

Warning: Whitelisting external resources opens potential security risks in your OTRS application! Only allow those resources that you are sure are not malicious and come from reputable sources. Keep in mind that if something is secure today, does not mean it will be tomorrow. Stay safe!

7.4.2 Cookies and Local Storage

The applications need cookies and local storage for proper working. Both the cookie and the local storage are stored on the user’s computer. This storage does not include personal data, they are only needed for the operation of the application.

The only cookie stored when the user visits the external interface is `AuthenticationCustomer`. This cookie stores the session ID of the logged in customer user and has a lifetime of 16 hours.

Additionally, the following key-value pairs are stored in the web browser’s local storage:

```
Plugins/Store/PublicChat/External/MutationsSharerStorage = {
    "type": "Plugins/Store/PublicChat/External/publicUuid",
    "payload": {
        "publicUuid": "<uuid-string>"
    }
}
Plugins/Store/LoggerOptions/External = {
    "logLevel": "info",
    "logRecord": false
}
Plugins/Store/PublicChat/External/MutationsSharerNotification = notification-
↪<notification-id>
Apps/External/Plugins/Store = {
    "accessToken": null,
    "deviceUuid": null,
    "isLanguageSet": false,
    "language": "en",
    "passwordExpirySoonWarning": null
}
Apps/External/Plugins/Store/MutationsSharerStorage = {
    "type": "language",
    "payload": {
        "language": "en"
    }
```

(continues on next page)

(continued from previous page)

```
}
Apps/External/Plugins/Store/PrivacyNotice = {
    "isNoticeVisible": true
}
Apps/External/Plugins/Store/MutationsSharerNotification = notification-
↪<notification-id>
Plugins/Store/PerlProfiler/External = {
    "perlProfiler": {}
}
Plugins/Store/PublicChat/External = {
    "publicUuid": "<uuid-string>"
}
```

The UUID values in the local storage are auto-generated when the user visits the external interface for the first time. The local storage has no expiration date and stores data used for the application itself. Additional notification entries might get spawned during the usage, and are used to sync the data state across multiple instances of the application.

CHAPTER 8

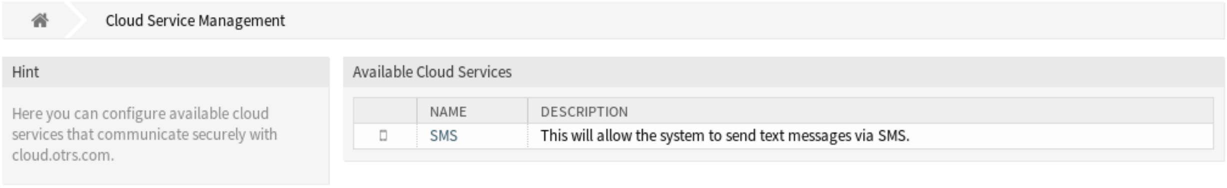
OTRS Group Services

A strong partner is good to have when dealing with mission-critical systems. Whether it be maintenance or for consuming cloud services, your partner should be tightly integrated.

The following chapter describes the tools you have available to integrate your system with the powerful cloud service offered by the *OTRS Group*.

8.1 Cloud Services

Use this screen to add cloud services to the system. A fresh OTRS installation doesn’t contain any configured cloud services by default. The cloud service management screen is available in the *Cloud Services* module of the *OTRS Group Services* group.



Cloud Service Management

Hint

Here you can configure available cloud services that communicate securely with cloud.otrs.com.

Available Cloud Services

	NAME	DESCRIPTION
□	SMS	This will allow the system to send text messages via SMS.

Fig. 1: Cloud Service Management Screen

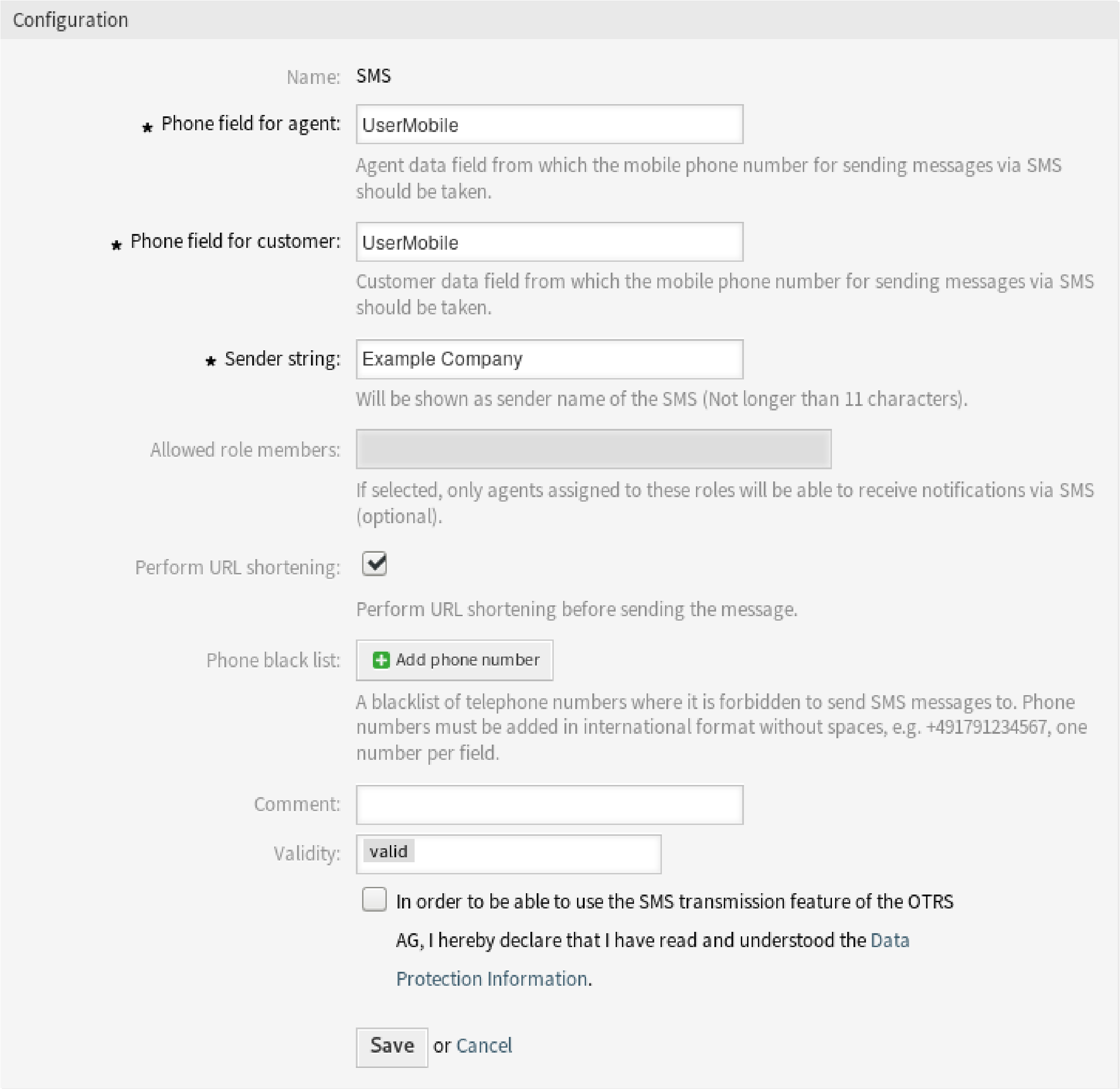
8.1.1 Manage Cloud Services

Activate SMS Cloud Service

To be able to use SMS cloud service in OTRS, you have to activate it first. To activate the SMS cloud service:

1. Click on the *Activate SMS Cloud Service* button in the left sidebar.

2. Fill in the required fields.
3. Click on the *Save* button.



Configuration

Name: SMS

★ Phone field for agent: UserMobile

Agent data field from which the mobile phone number for sending messages via SMS should be taken.

★ Phone field for customer: UserMobile

Customer data field from which the mobile phone number for sending messages via SMS should be taken.

★ Sender string: Example Company

Will be shown as sender name of the SMS (Not longer than 11 characters).

Allowed role members:

If selected, only agents assigned to these roles will be able to receive notifications via SMS (optional).

Perform URL shortening: ☑

Perform URL shortening before sending the message.

Phone black list:

A blacklist of telephone numbers where it is forbidden to send SMS messages to. Phone numbers must be added in international format without spaces, e.g. +491791234567, one number per field.

Comment:

Validity: valid

□ In order to be able to use the SMS transmission feature of the OTRS AG, I hereby declare that I have read and understood the Data Protection Information.

Fig. 2: Add Cloud Service Screen

Configuration

The following settings are available when adding an SMS cloud service. The fields marked with an asterisk are mandatory.

Phone field for agent * Agent data field from which the mobile phone number for sending messages via SMS should be taken.

Phone field for customer * Customer data field from which the mobile phone number for sending messages via SMS should be taken.

Sender string * Will be shown as sender name of the SMS (Not longer than 11 characters).

Allowed role members If selected, only agents assigned to these roles will be able to receive notifications via SMS.

Perform URL shortening Perform URL shortening before sending the message.

Phone black list A blacklist of telephone numbers where it is forbidden to send SMS messages to. Phone numbers must be added in international format without spaces, e.g. +491791234567, one number per field.

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

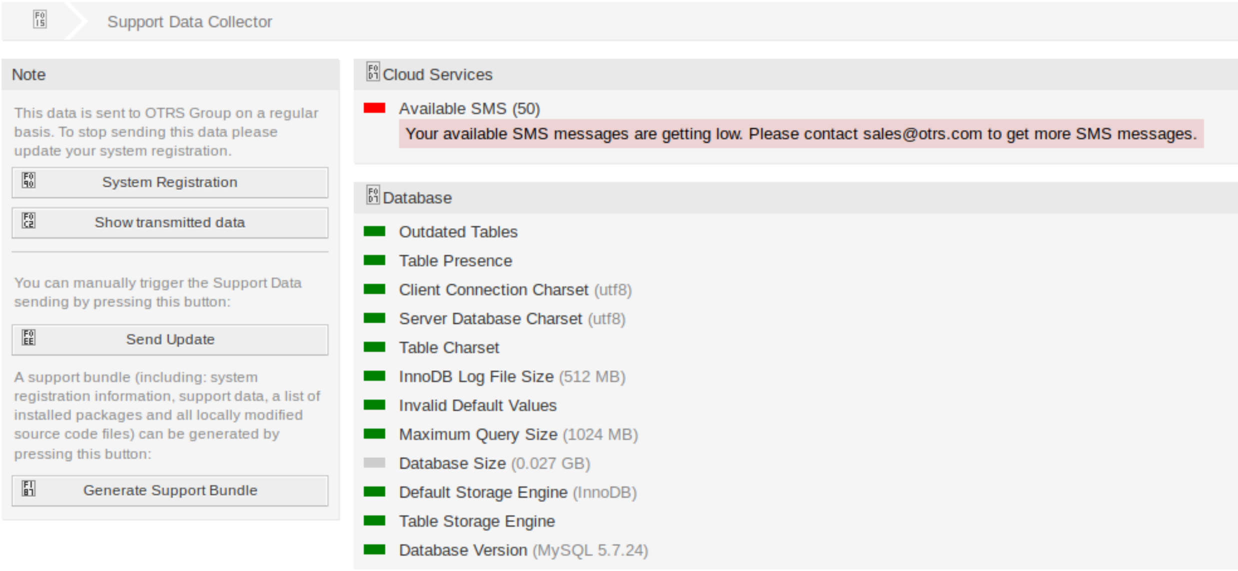
Validity Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.

Data Protection Information In order to be able to use the SMS transmission feature of the OTRS AG, the Data Protection Information needs to be read and understood.

8.2 Support Data Collector

Support data collector is used to collect some data and sent to OTRS Group on a regular basis, if the system is registered.

Use this screen to review the data to be sent to OTRS Group. The support data collector screen is available in the *Support Data Collector* module of the *OTRS Group Services* group.



This data is sent to OTRS Group on a regular basis. To stop sending this data please update your system registration.

You can manually trigger the Support Data sending by pressing this button:

A support bundle (including: system registration information, support data, a list of installed packages and all locally modified source code files) can be generated by pressing this button:

🟥 Available SMS (50)

Your available SMS messages are getting low. Please contact sales@otrs.com to get more SMS messages.

🟩 Outdated Tables
🟩 Table Presence
🟩 Client Connection Charset (utf8)
🟩 Server Database Charset (utf8)
🟩 Table Charset
🟩 InnoDB Log File Size (512 MB)
🟩 Invalid Default Values
🟩 Maximum Query Size (1024 MB)
⬜ Database Size (0.027 GB)
🟩 Default Storage Engine (InnoDB)
🟩 Table Storage Engine
🟩 Database Version (MySQL 5.7.24)

Fig. 3: Download Support Bundle Screen

8.2.1 Manage Support Data Collector

Support data collector is used to collect some data and sent to OTRS Group on a regular basis, if the system is registered. To register your system:

1. Click on the *System Registration* button in the left sidebar.
2. Follow the registration instructions.

To show what kind of data will be sent:

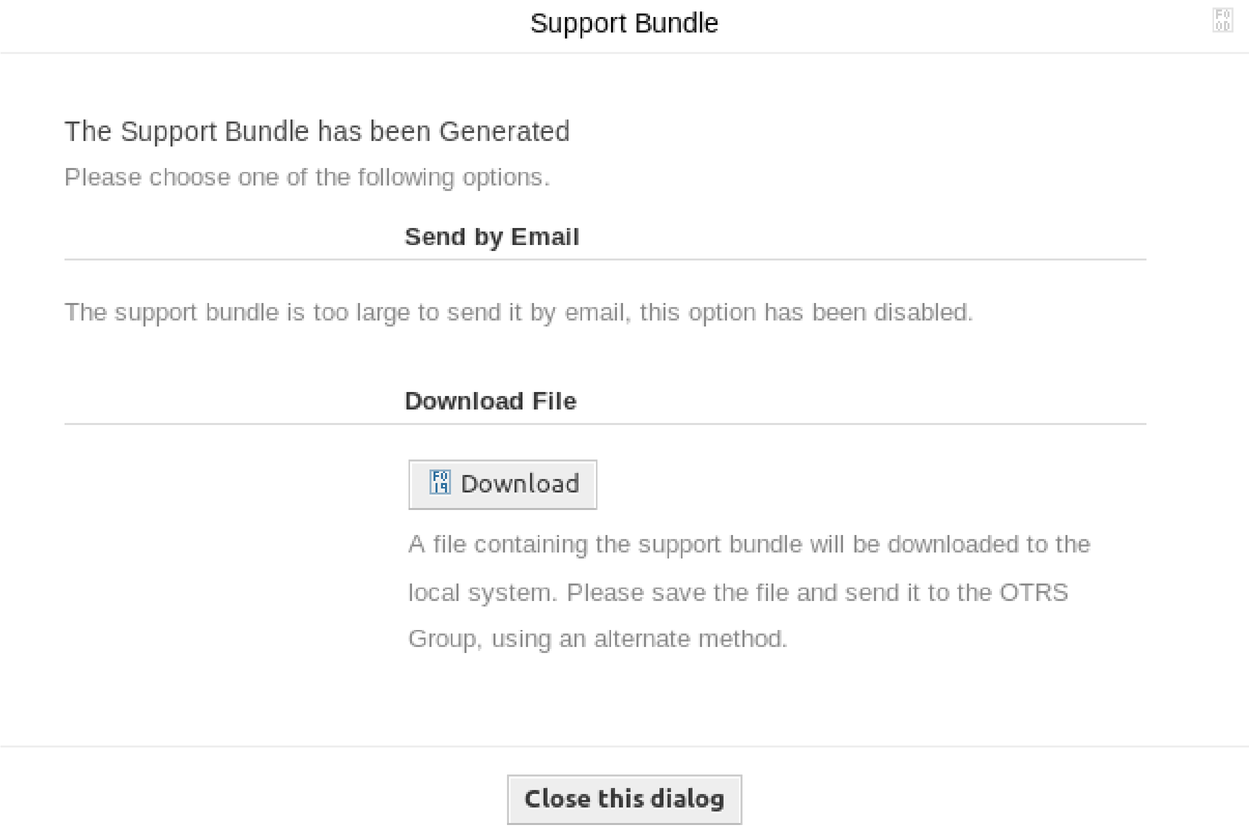
1. Click on the *Show transmitted data* button in the left sidebar.
2. Review the *System Registration Data* and *Support Data* in the newly opened screen.

To manually trigger the support data sending:

1. Click on the *Send Update* button in the left sidebar.

To generate a support bundle:

1. Click on the *Generate Support Bundle* button in the left sidebar.
2. Download the generated support bundle.
3. Open it with an archive manager and review the content.



Support Bundle

The Support Bundle has been Generated

Please choose one of the following options.

Send by Email

The support bundle is too large to send it by email, this option has been disabled.

Download File

A file containing the support bundle will be downloaded to the local system. Please save the file and send it to the OTRS Group, using an alternate method.

Fig. 4: Download Support Bundle Dialog

8.2.2 Collected Data

The screen contains several sections. Each section has some entries with a traffic light, that indicates the following:

- Gray LED means information, just displays a value.
- Green LED means OK, the entry has a good value.
- Yellow LED means notification, you have to check the value, but it is not an error.
- Red LED means error, you have to do something to solve the issue.

Cloud Services Section

This section displays information about OTRS cloud services.

Available SMS This entry shows information about your available SMS messages. If they are getting low, the LED changes to red.

Database Section

This section displays information about the database used by OTRS.

Outdated Tables Display the outdated database tables. Green LED means, there are no outdated tables.

Table Presence Display whether all needed tables exist in the database or not.

Client Connection Charset Display the character set for the client connection. It must be `utf8`.

Server Database Charset Display the character set of the database server. It must be `utf8`.

Table Charset Display the character set of the database table. It must be `utf8`.

InnoDB Log File Size Display the log file size for InnoDB driver. It must be at least `512 MB`.

Invalid Default Values Display the invalid default values. Green LED means, there are no invalid default values.

Maximum Query Size Display the maximum size of a database query. It must be at least `1024 MB`.

Database Size Display the size of database. This is just an information.

Default Storage Engine Display the default storage engine of the database. It must be `InnoDB`.

Table Storage Engine Display the storage engine of the database tables. It must be `InnoDB`.

Database Version Display the database driver version. Green LED means, the version is high enough.

Document Search Section

This section displays information about document search and the used cluster.

Cluster The name of the used cluster.

Cluster Health Details Display some internal variables of the used cluster.

Indices Health Display information about indices.

Indices Size Display the size of each index.

Node Health Display information about the used node.

Operating System Section

This section displays information about the running operating system and installed software components.

Environment Dependencies Display information about environment dependencies.

OTRS Disk Partition Display the disk partition to where OTRS is installed.

Information Disk Partitions Usage Display the used space per disk partitions.

Distribution Display the distribution name of the operating system.

Kernel Version Display the kernel version of the operating system.

System Load Display the system load of the operating system. The system load should be at maximum the number of CPUs the system has (e.g. a load of 8 or less on a system with 8 CPUs is OK).

Perl Version Display the version of Perl.

Free Swap Space (%) Display the free swap space as percentages. There should be more than 60% free swap space.

Used Swap Space (MB) Display the used swap space in megabytes. There should be no more than 200 MB swap space used.

OTRS Section

This section displays information about the OTRS instance.

Article Search Index Status Display information about indexed articles.

Articles Per Communication Channel Display the number of articles per communication channels.

Communication Log Display aggregated information about communications.

Communication Log Account Status (last 24 hours) Display information about communication log account status in the last 24 hours.

Concurrent Users Details Display information about the logged in users at the same time separated by hourly.

Concurrent Users Display information about the number of maximum logged in users in the same time.

Config Settings Display some important configuration settings from system configurations.

Daemon Display whether the OTRS daemon is running or not.

Database Records Display the main OTRS object and the related number of records in the database.

Default Admin Password Green LED means, that the default administrator password was changed.

Email Sending Queue Display the number of emails that are queued for sending.

FQDN (Domain Name) Display the fully qualified domain name set in system configuration setting FQDN.

File System Writable Display whether the file system is writable or not.

Legacy Configuration Backups Green LED means, there are no legacy configuration backup files found.

Package Installation Status Green LED means, that all packages are installed correctly.

Package Framework Version Status Green LED means, that the OTRS framework version is suitable for the installed packages.

Package Verification Status Green LED means, that all installed packages are verified by the OTRS Group.

Package List Display the list of installed packages.

Push Events Status Display the status of the push events per day.

Session Config Settings Display the maximum allowed sessions per agents and customers.

Spooled Emails Display the number of emails that are in the sending pool.

SystemID Display the system identifier set in system configuration setting SystemID.

Invalid Users with Locked Tickets Display the number of users, who are set to invalid, but have some ticket locked for him.

Open Tickets Display the number of open tickets in the system. You will not have performance trouble until you have about 60,000 open tickets in your system.

Ticket Search Index Module Display the ticket search index module set in system configuration setting Ticket::SearchIndex::ForceUnfilteredStorage.

Time Settings Display timezone information for OTRS, for the calendars and for users.

UI - Agent Skin Usage Display the used skins per agents.

UI - Agent Theme Usage Display the used theme on the agent interface.

UI - Special Statistics Display some statistics about personal modifications like using favorites, custom menu ordering, etc.

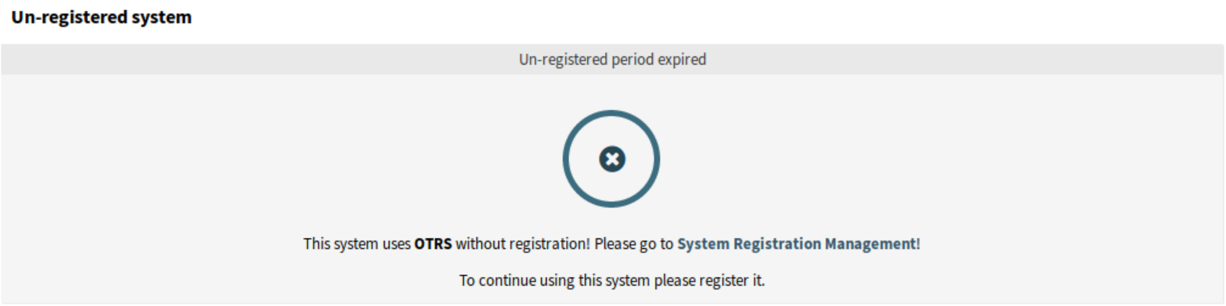
OTRS Version Display the version number of OTRS.

WebSocket Connection Display connection status for WebSocket connections.

8.3 System Registration

If the free trial or the registration period is expired, the system will lock itself. In these cases you have to register your system or you have to extend the registration.

Use this screen to register your system with the OTRS Group. The registration screen is available in the *System Registration* module of the *OTRS Group Services* group.



Un-registered system

Un-registered period expired

This system uses **OTRS** without registration! Please go to **System Registration Management!**

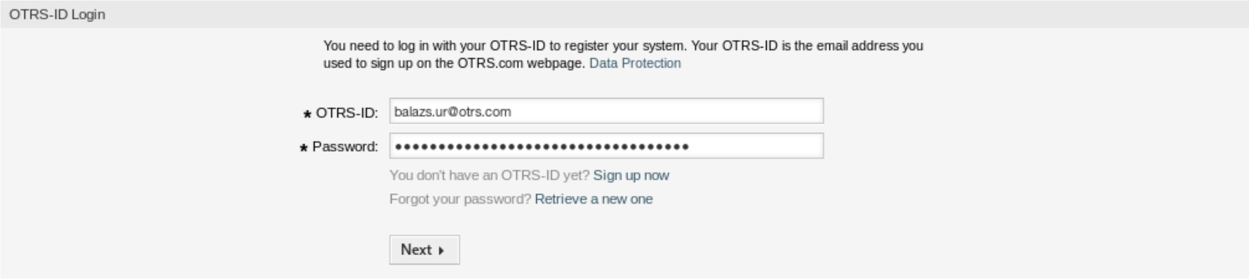
To continue using this system please register it.

Fig. 5: Un-Registred System Screen

8.3.1 Manage System Registration

To register your system:

1. Obtain an OTRS ID. You have to register in the OTRS Portal. After the registration you will get your OTRS ID.
2. Go to the *System Registration* module of the *OTRS Group Services* group, and enter your OTRS ID and your password.



OTRS-ID Login

You need to log in with your OTRS-ID to register your system. Your OTRS-ID is the email address you used to sign up on the OTRS.com webpage. Data Protection

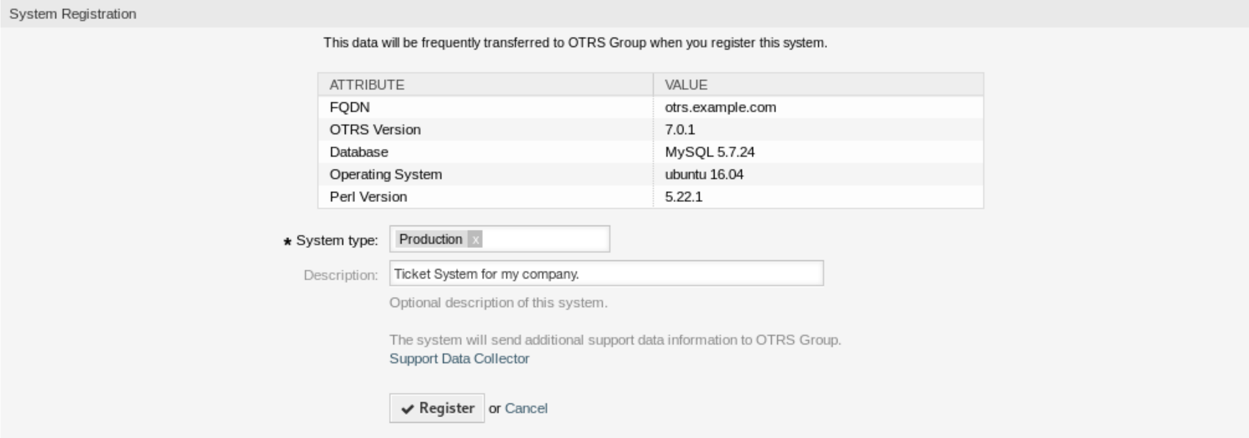
⋆ OTRS-ID:	balazs.ur@otrs.com
⋆ Password:	••••••••••••••••••••••••••••••••••

You don't have an OTRS-ID yet? Sign up now

Forgot your password? Retrieve a new one

Fig. 6: System Registration - Add OTRS ID

3. Click on the *Next* button.
4. Select the system type and enter a description.



System Registration

This data will be frequently transferred to OTRS Group when you register this system.

ATTRIBUTE	VALUE
FQDN	otrs.example.com
OTRS Version	7.0.1
Database	MySQL 5.7.24
Operating System	ubuntu 16.04
Perl Version	5.22.1

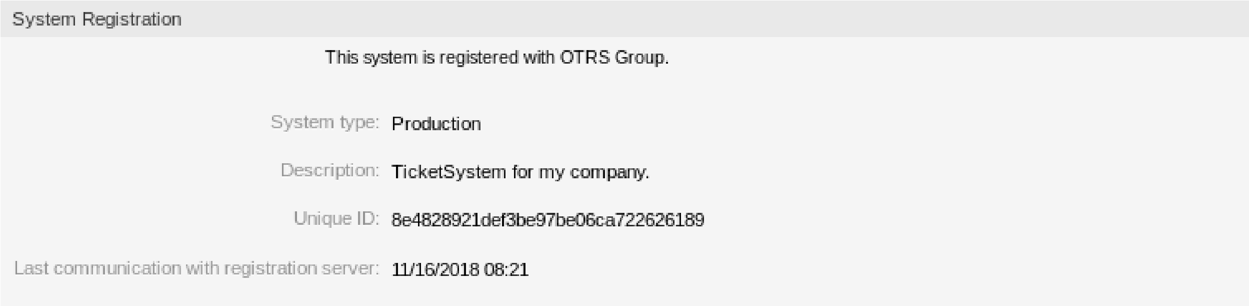
⋆ System type:	Production x
Description:	Ticket System for my company.

Optional description of this system.

The system will send additional support data information to OTRS Group.
Support Data Collector

Fig. 7: System Registration - Select System Type

5. Click on the *Register* button.
6. Check your registration.



System Registration

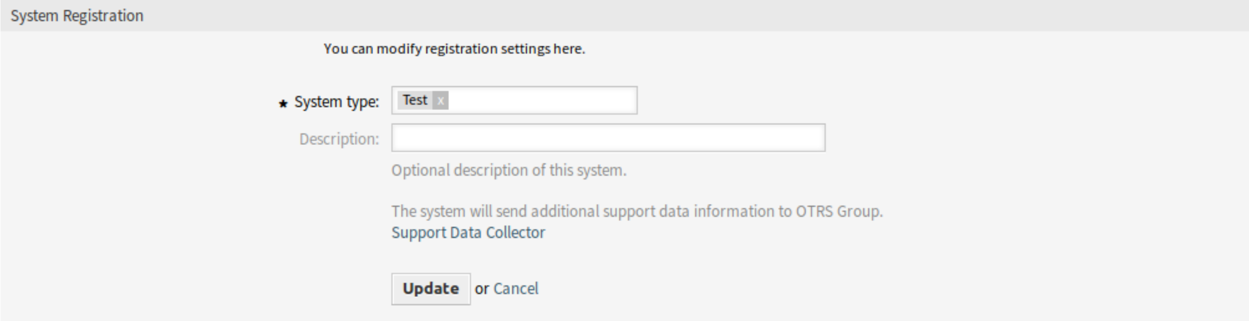
This system is registered with OTRS Group.

System type:	Production
Description:	TicketSystem for my company.
Unique ID:	8e4828921def3be97be06ca722626189
Last communication with registration server:	11/16/2018 08:21

Fig. 8: Registered System

To edit the system registration:

1. Click on the *Edit details* button in the left sidebar.
2. Modify the system type and the description.
3. Click on the *Update* button.



System Registration

You can modify registration settings here.

★ System type: Test ×

Description:

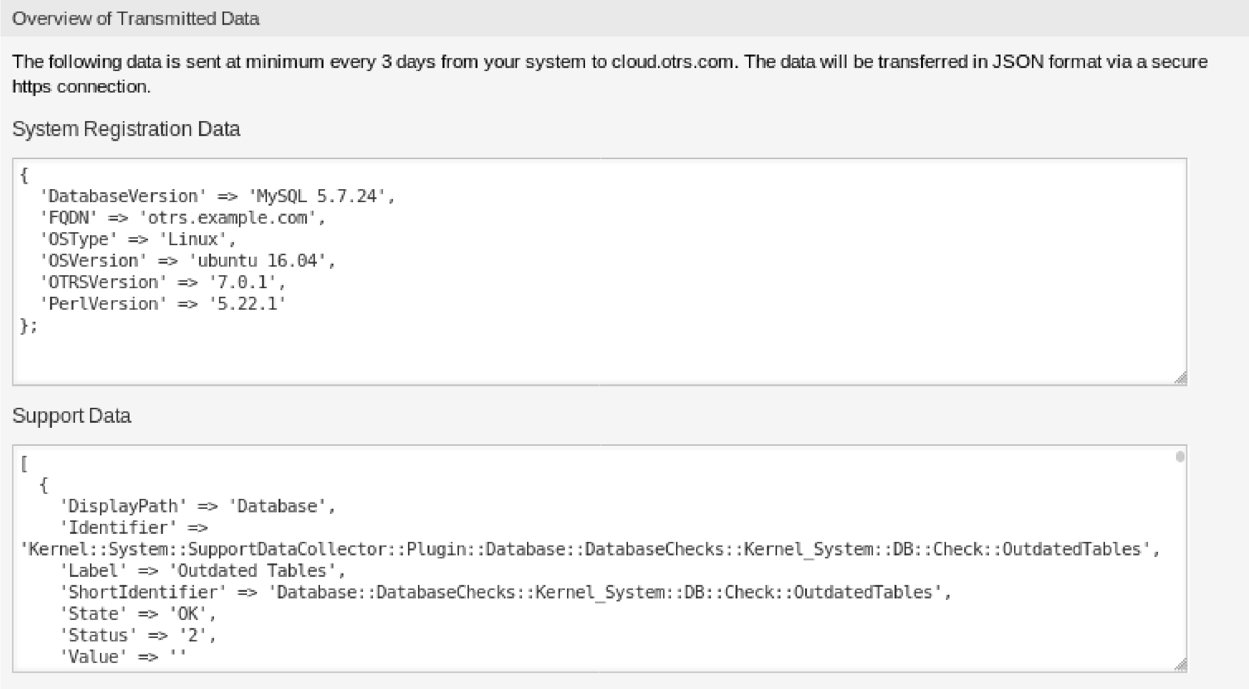
Optional description of this system.

The system will send additional support data information to OTRS Group.
Support Data Collector

Fig. 9: Edit System Registration Screen

To show the transmitted data:

1. Click on the *Show transmitted data* button in the left sidebar.
2. Review the data in JSON format.



Overview of Transmitted Data

The following data is sent at minimum every 3 days from your system to cloud.otrs.com. The data will be transferred in JSON format via a secure https connection.

System Registration Data

```
{
  'DatabaseVersion' => 'MySQL 5.7.24',
  'FQDN' => 'otrs.example.com',
  'OSType' => 'Linux',
  'OSVersion' => 'ubuntu 16.04',
  'OTRSVersion' => '7.0.1',
  'PerlVersion' => '5.22.1'
};
```

Support Data

```
[
  {
    'DisplayPath' => 'Database',
    'Identifier' =>
'Kernel::System::SupportDataCollector::Plugin::Database::DatabaseChecks::Kernel_System::DB::Check::OutdatedTables',
    'Label' => 'Outdated Tables',
    'ShortIdentifier' => 'Database::DatabaseChecks::Kernel_System::DB::Check::OutdatedTables',
    'State' => 'OK',
    'Status' => '2',
    'Value' => ''
```

Fig. 10: Overview of Transmitted Data Screen

See also:

Detailed explanation of the fields is available in the *Support Data Collector* chapter.

To overview the registered system:

1. Click on the *Overview of registered systems* button in the left sidebar.
2. Log in to the OTRS Portal.

3. Review the list of registered systems.

To deregister a system:

1. Click on the *Deregister system* button in the left sidebar.

Note: You can’t deregister your system if you’re using the **STORM powered by OTRS™** or having a valid service contract.

CHAPTER 9

Chat

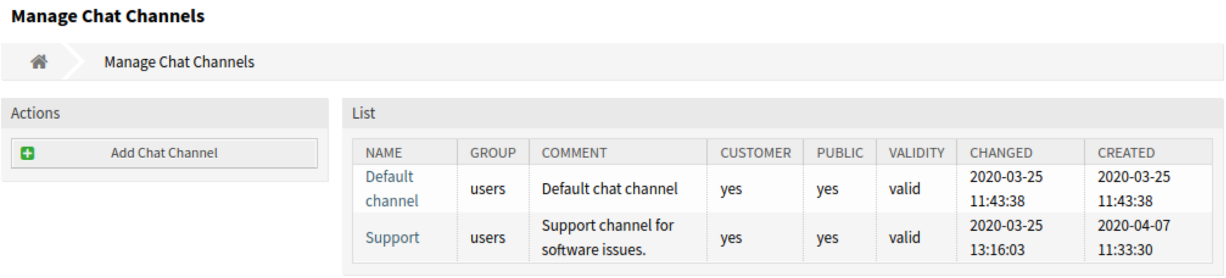
The chat feature allows the agents to provide real time communication with each other, with customer users or with public users. The chat feature is enabled by default.

9.1 Chat Channels

Being able to offer chat possibilities to customers is a must-have for many organizations. Depending on the amount of customer chat requests and the organization’ s structure, it must be possible to group chat requests.

OTRS offers chat channels with different permissions per channel, so it is, e.g. possible to have different chat channels for registered contract customers and public prospects.

Use this screen to add chat channels to the system. A fresh OTRS installation already contains a chat channel by default. The chat channel management screen is available in the *Chat Channels* module of the *Chat* group.



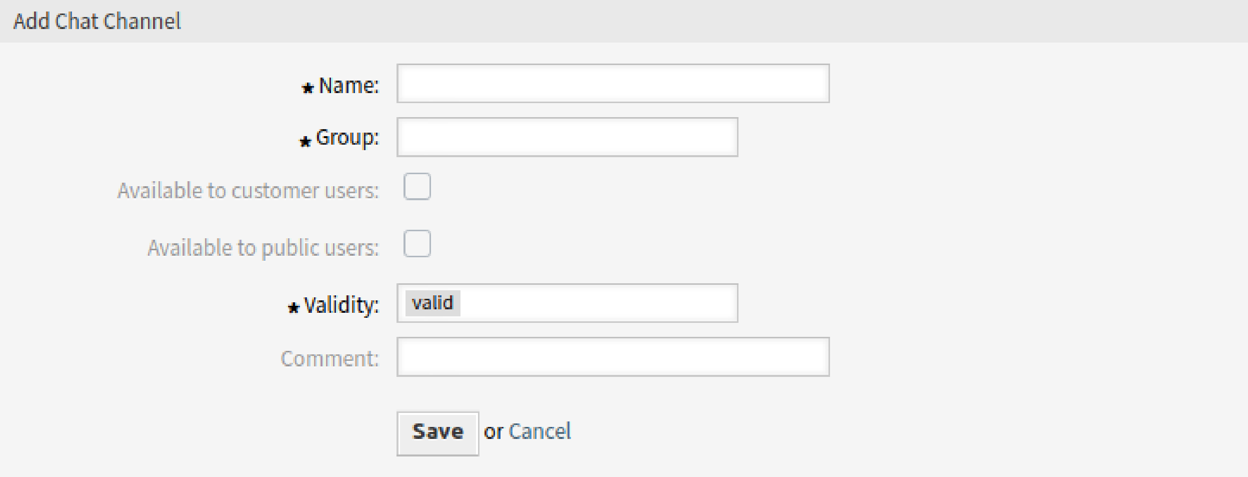
NAME	GROUP	COMMENT	CUSTOMER	PUBLIC	VALIDITY	CHANGED	CREATED
Default channel	users	Default chat channel	yes	yes	valid	2020-03-25 11:43:38	2020-03-25 11:43:38
Support	users	Support channel for software issues.	yes	yes	valid	2020-03-25 13:16:03	2020-04-07 11:33:30

Fig. 1: Chat Channel Management Screen

9.1.1 Manage Chat Channels

To add a chat channel:

1. Click on the *Add Chat Channel* button in the left sidebar.
2. Fill in the required fields.
3. Click on the *Save* button.



Add Chat Channel

⋆ Name:

⋆ Group:

Available to customer users: □

Available to public users: □

⋆ Validity: valid

Comment:

Fig. 2: Add Chat Channel Screen

> **Warning:** Chat channels can not be deleted from the system. They can only be deactivated by setting the *Validity* option to *invalid* or *invalid-temporarily*.

To edit a chat channel:

1. Click on a chat channel in the list of chat channels.
2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.

9.1.2 Chat Channel Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

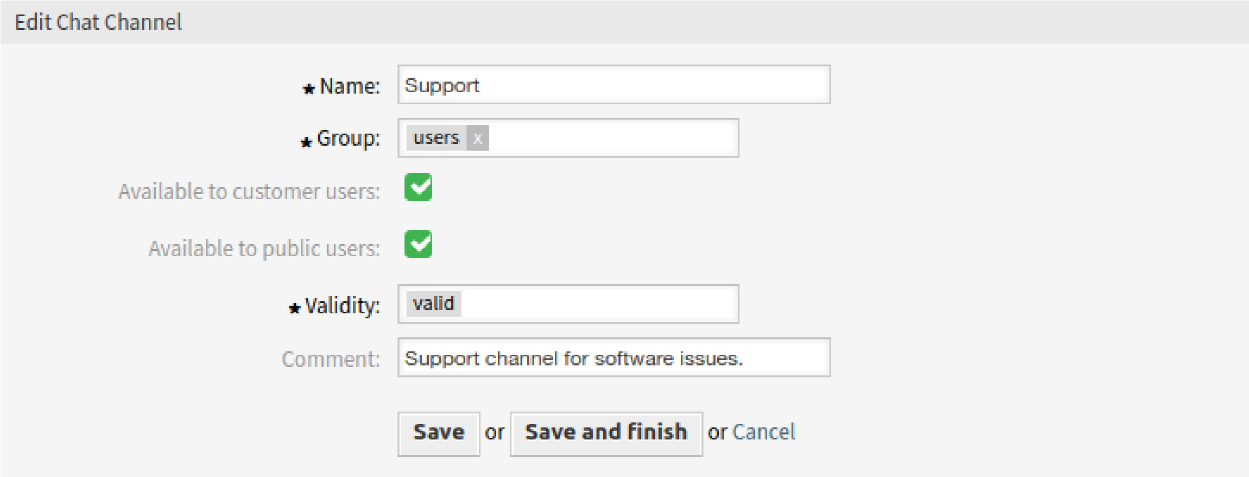
Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Group * Select which *Groups* can access the chat channel.

Available to customer users Select the checkbox if you want to display the chat channel for customer users.

Available to public users Select the checkbox if you want to display the chat channel for public users.

Validity * Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.



Edit Chat Channel

★ Name:	Support
★ Group:	users x
Available to customer users:	☑
Available to public users:	☑
★ Validity:	valid
Comment:	Support channel for software issues.

Fig. 3: Edit Chat Channel Screen

Comment Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

9.1.3 Manage Chat Support

Note: The chat feature is enabled by default.

To enable or disable the chat feature:

1. Go to *System Configuration* screen.
2. Navigate to *Core* → *Chat* in the navigation tree.
3. Enable or disable the setting `ChatEngine::Active`.

When the chat feature is enabled, the *Default channel* is automatically created if it does not exist.

To review all chat related settings:

1. Go to *System Configuration* screen.
2. Search for the term `ChatEngine` in the search box.
3. Review the settings.

See also:

Agents need to have at least *ro* permissions to the groups set up in the following settings:

- `ChatEngine::PermissionGroup::ChatReceivingAgents`
- `ChatEngine::PermissionGroup::ChatStartingAgents`

9.2 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 modify the fields in the configuration widget. 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.

Use this screen to generate the chat integration code. The chat integration configuration and snippet generator is available in the *Chat Integration* module of the *Chat* group.

To configure the chat and generate code snippet:

1. Select a pre-selected chat channel in the *Pre-selected channel* field.
2. Select a primary color for the chat design.
3. Customize the texts in the *Texts* section.

   Note: All changes made in the *Configuration* tab will not be saved. Every time you open this dialog, all fields will be reset to default values.

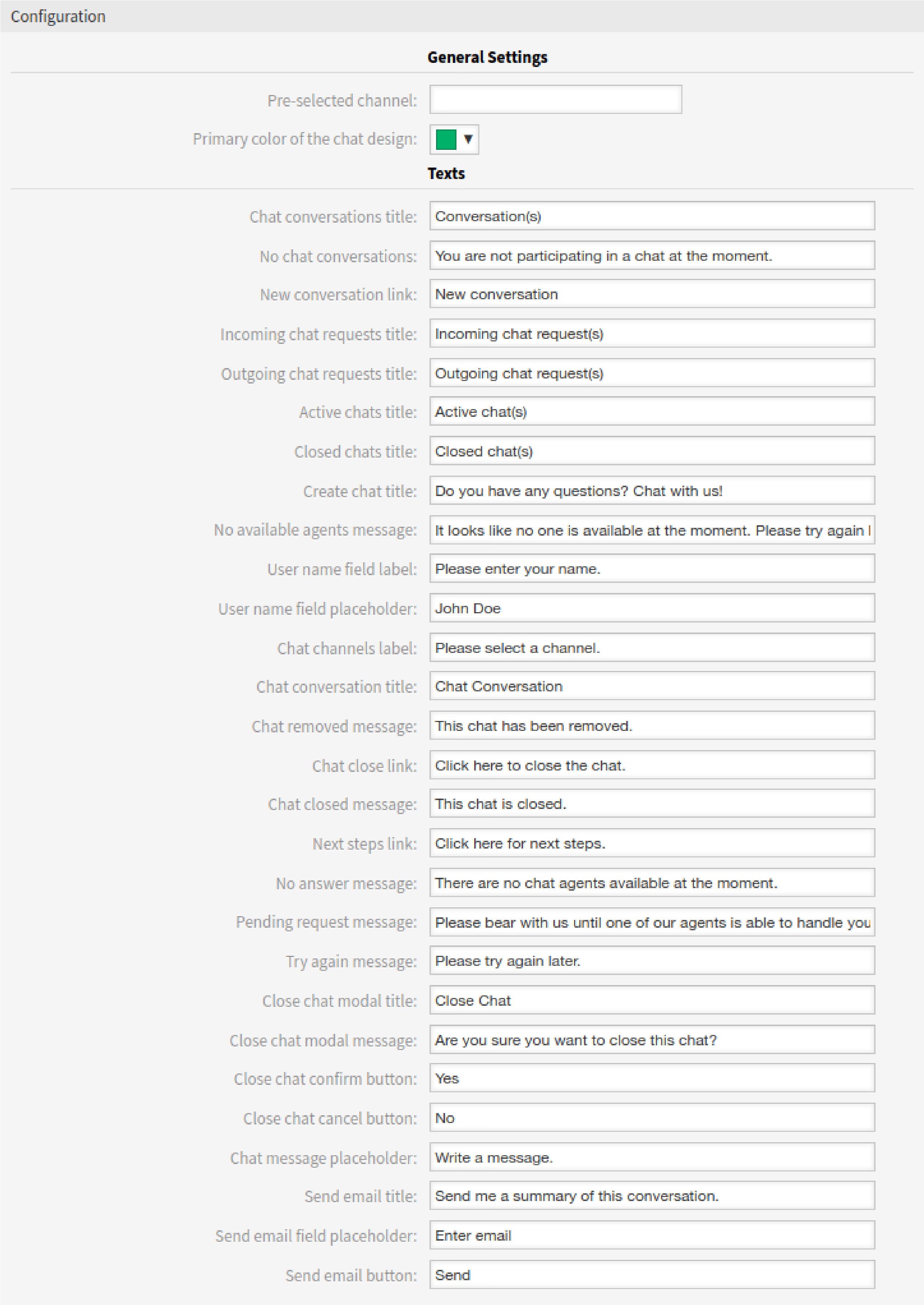
4. You can check the result in the *Preview* widget.

   Note: The preview uses the real chat module. Other agents have to be available for chatting to preview all features.

5. Copy the code snippet from the *Integration Code* widget and paste it into your website right before the `</body>` element.

If mixed content warning is displayed in the browser console, an administrator has to check that the system configuration setting `HttpType` is properly set. The website must run on the same protocol for chat widget to work.

For example, if the website is running OTRS on SSL, the system configuration option must be set to `https`.



Configuration

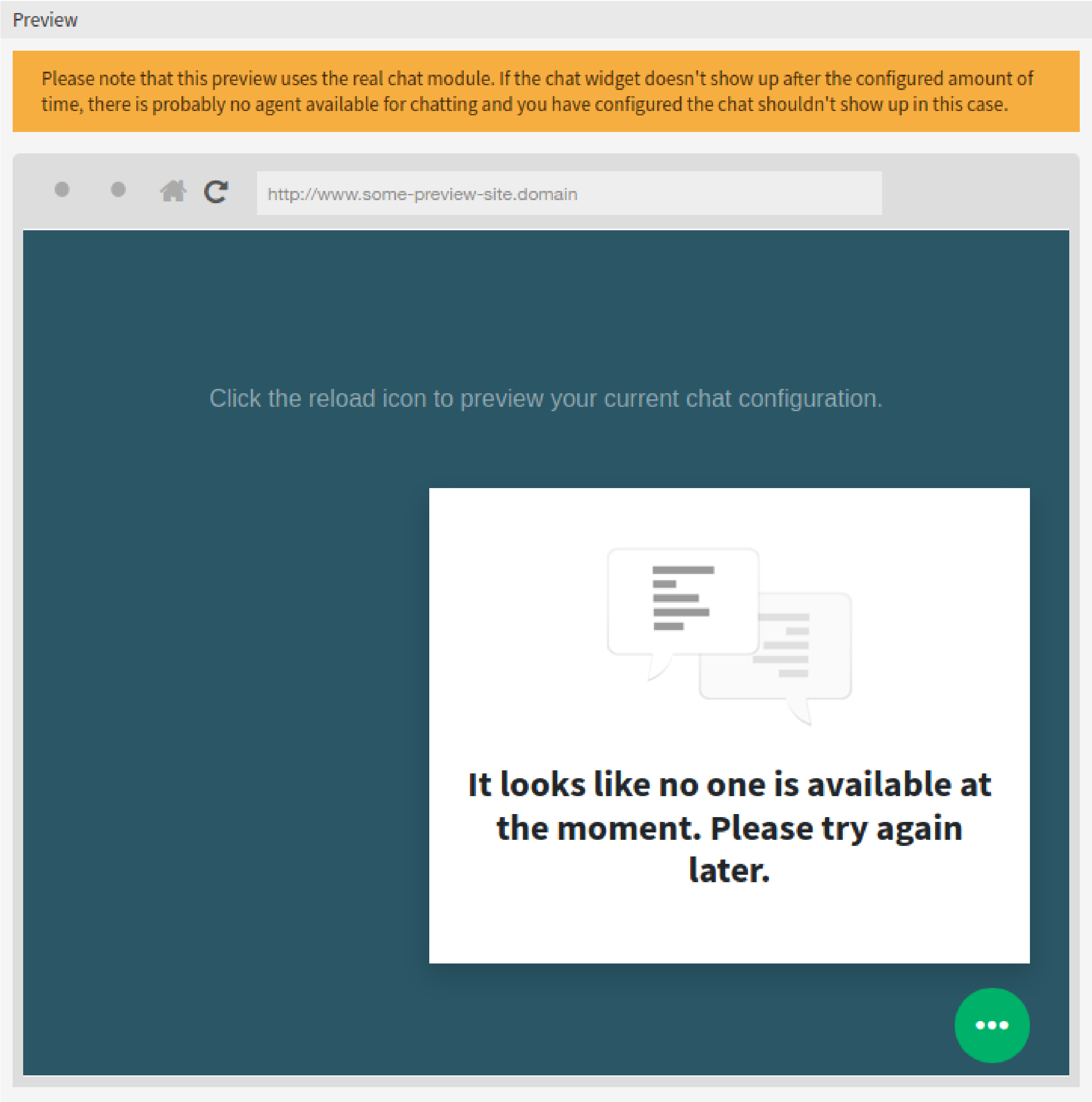
General Settings

Pre-selected channel:	
Primary color of the chat design:	🟩 ▼

Texts

Chat conversations title:	Conversation(s)
No chat conversations:	You are not participating in a chat at the moment.
New conversation link:	New conversation
Incoming chat requests title:	Incoming chat request(s)
Outgoing chat requests title:	Outgoing chat request(s)
Active chats title:	Active chat(s)
Closed chats title:	Closed chat(s)
Create chat title:	Do you have any questions? Chat with us!
No available agents message:	It looks like no one is available at the moment. Please try again l
User name field label:	Please enter your name.
User name field placeholder:	John Doe
Chat channels label:	Please select a channel.
Chat conversation title:	Chat Conversation
Chat removed message:	This chat has been removed.
Chat close link:	Click here to close the chat.
Chat closed message:	This chat is closed.
Next steps link:	Click here for next steps.
No answer message:	There are no chat agents available at the moment.
Pending request message:	Please bear with us until one of our agents is able to handle you
Try again message:	Please try again later.
Close chat modal title:	Close Chat
Close chat modal message:	Are you sure you want to close this chat?
Close chat confirm button:	Yes
Close chat cancel button:	No
Chat message placeholder:	Write a message.
Send email title:	Send me a summary of this conversation.
Send email field placeholder:	Enter email
Send email button:	Send

Fig. 4: Chat Integration – Configuration



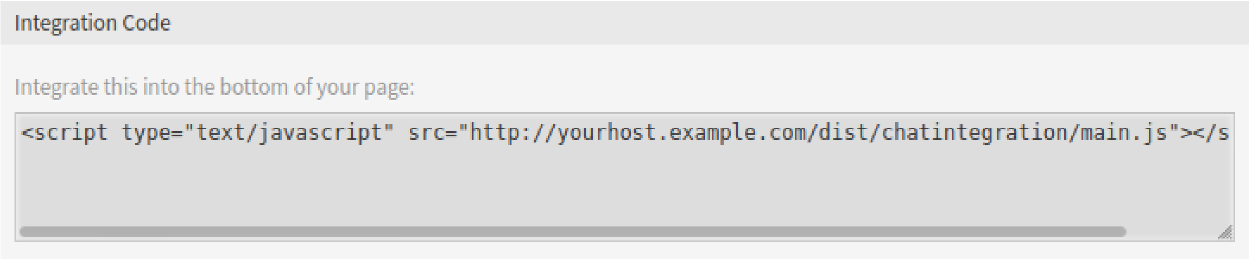
Please note that this preview uses the real chat module. If the chat widget doesn't show up after the configured amount of time, there is probably no agent available for chatting and you have configured the chat shouldn't show up in this case.

http://www.some-preview-site.domain

Click the reload icon to preview your current chat configuration.

It looks like no one is available at the moment. Please try again later.

Fig. 5: Chat Integration – Preview



Integrate this into the bottom of your page:

```
<script type="text/javascript" src="http://yourhost.example.com/dist/chatintegration/main.js"></s
```

Fig. 6: Chat Integration – Integration Code

CHAPTER 10

Administration

Any system requires configuration. Configuring a system should be an easy task and the tools for configuration fit-for-purpose.

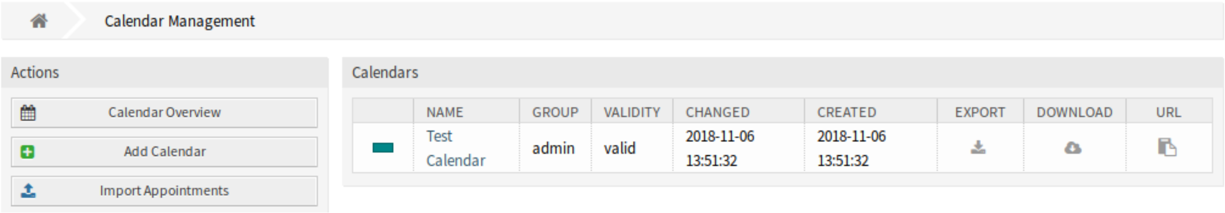
OTRS offers several administration tools to configure, monitor, control and extend OTRS.

10.1 Calendars

When working with customers, resource planning and scheduling can be a complex task. Appointments enable you to meet your customers where and whenever needed.

OTRS supports this requirement with calendars. Calendars allow management of appointments and resources inside the ticket system. You can connect your tickets to scheduled tasks and make them available to all users to see. This feature adds transparency to show your teams workload and prevent users from promising resources which are not available.

Use this screen to manage calendars in the system. A fresh OTRS installation contains no calendars by default. The calendar management screen is available in the *Calendars* module of the *Administration* group.



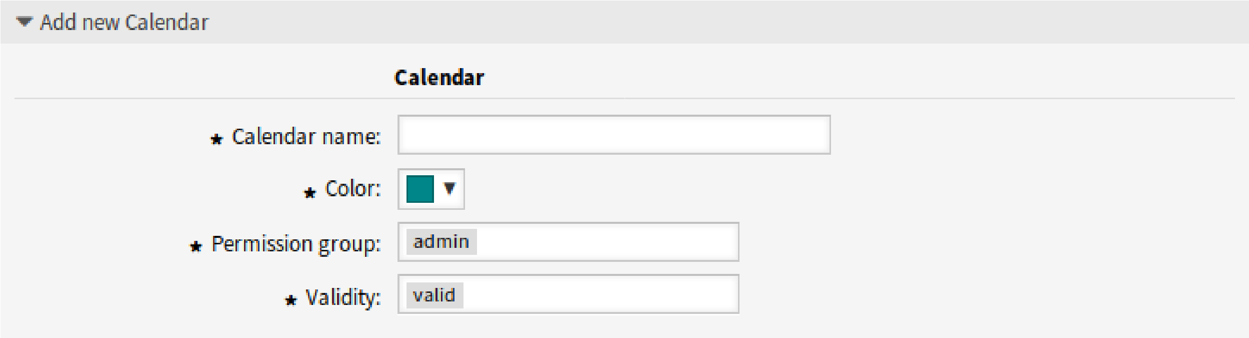
	NAME	GROUP	VALIDITY	CHANGED	CREATED	EXPORT	DOWNLOAD	URL
🟦	Test Calendar	admin	valid	2018-11-06 13:51:32	2018-11-06 13:51:32	⭳	☁	⎘

Fig. 1: Calendar Management Screen

10.1.1 Manage Calendars

To add a new calendar:

1. Click on the *Add Calendar* button in the left sidebar.
2. Fill in the required fields.
3. Click on the *Save* button.



Add new Calendar

Calendar

* Calendar name:

* Color: 🟦 ▼

* Permission group: admin

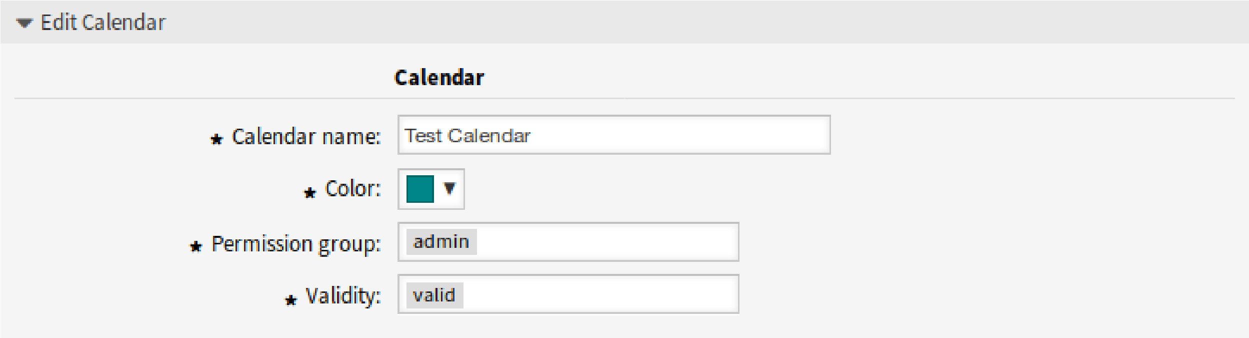
* Validity: valid

Fig. 2: Add New Calendar Screen

> **Warning:** Calendars can not be deleted from the system. They can only be deactivated by setting the *Validity* option to *invalid* or *invalid-temporarily*.

To edit a calendar:

1. Click on a calendar in the list of calendars.
2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.



Edit Calendar

Calendar

* Calendar name: Test Calendar

* Color: 🟦 ▼

* Permission group: admin

* Validity: valid

Fig. 3: Edit Calendar Screen

To export a calendar:

1. Click on the export icon in the list of calendars.
2. Choose a location in your computer to save the `Export_Calendar_CalendarName.yml` file.

To import calendars:

1. Click on the *Browse*… button in the left sidebar.
2. Select a previously exported `.yml` file.
3. Click on the *Overwrite existing entities* checkbox, if you would like to overwrite the existing calendars.

4. Click on the *Import Calendar* button.

Note: If several calendars are added to the system, use the filter box to find a particular calendar by just typing the name to filter.

10.1.2 Calendar Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

General Calendar Settings

Calendar name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Color * The calendar color that will be displayed in the calendar overview screens.

To change the calendar color, just select a new color from the color palette. You can chose from the preselected colors or define other colors by choosing it from the color selector or typing the hexadecimal value.

Permission group * Select which *Groups* can access the calendar.

Depending on the group field, the system will allow users the access to the calendar according to their permission level.

- Read only: users can see and export all appointments in the calendar.
- Move into: users can modify appointments in the calendar, but without changing the calendar selection.
- Create: users can create and delete appointments in the calendar.
- Read/write: users can manage the calendar itself.

Validity * Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.

Calendar Ticket Appointments

Define rules for creating automatic appointments in this calendar based on ticket data. To add a new rule, click on the *Add Rule* button.

Start date Select a start date for the ticket appointment.

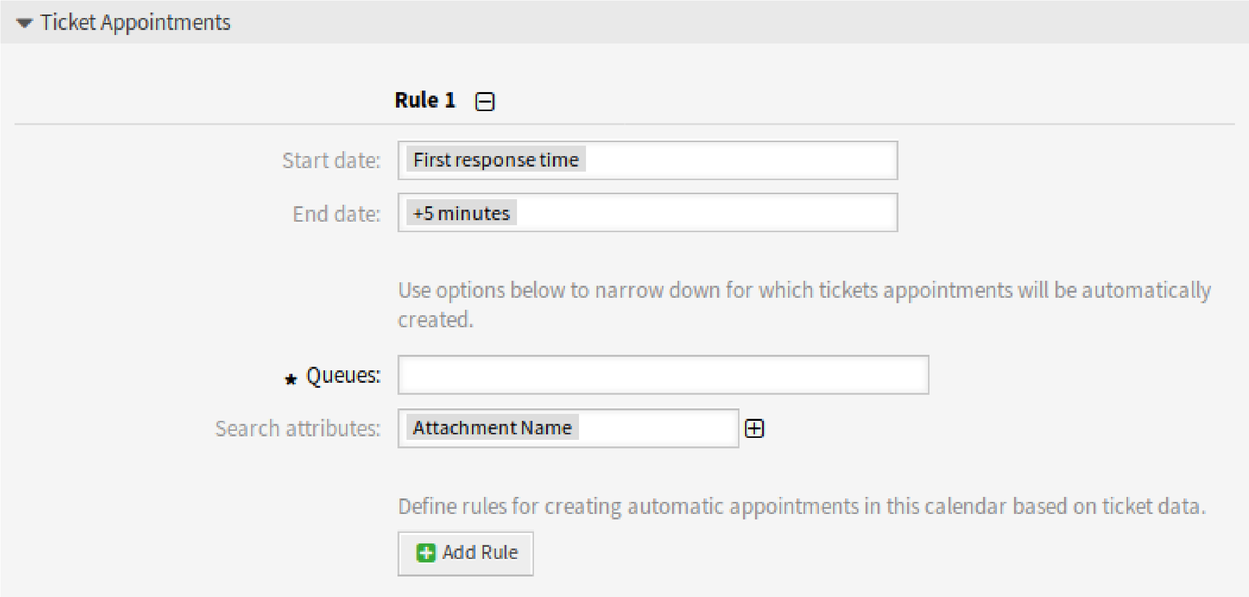
End date Select the end date for the ticket appointment.

Queues * Select one or more queues to narrow down for which tickets appointments will be automatically created.

Search attributes Additional search attributes can be added for further filtering by selecting an attribute and clicking on the button.

By default, these events can trigger an update of a calendar appointment:

- `TicketSLAUpdate`
- `TicketQueueUpdate`



▾ Ticket Appointments

Rule 1 ⊟

Start date: First response time

End date: +5 minutes

Use options below to narrow down for which tickets appointments will be automatically created.

★ Queues:

Search attributes: Attachment Name ⊞

Define rules for creating automatic appointments in this calendar based on ticket data.

Fig. 4: Calendar Settings - Ticket Appointments

- `TicketStateUpdate`
- `TicketCreate`
- `ArticleCreate`
- `TicketPendingTimeUpdate`
- `TicketDynamicFieldUpdate_.*`

10.1.3 Import Appointments

If at least one calendar have been added to the system, it is possible to import some appointments into the calendar.

To import some appointments:

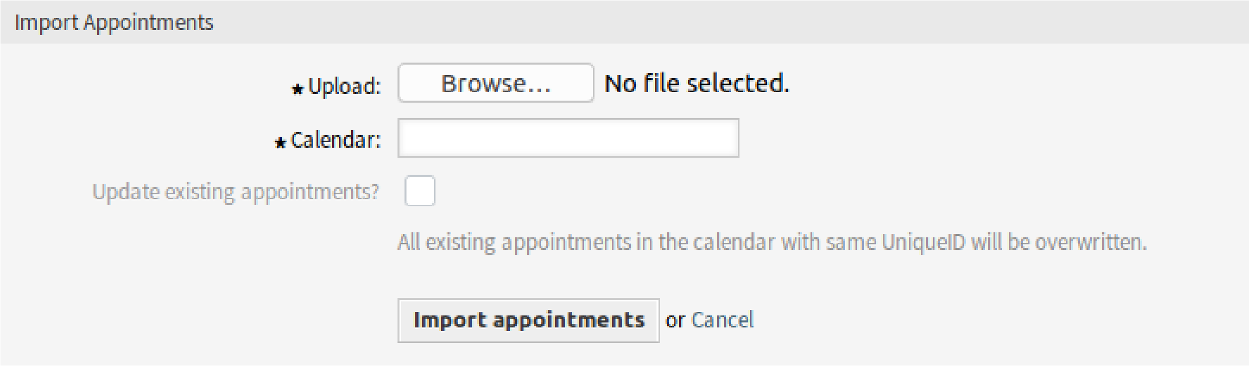
1. Click on the *Import Appointments* button in the left sidebar.
2. Upload an iCal file and select a calendar.
3. Click on the *Import appointments* button.

Upload * Click on the *Browse…* button, and select a valid iCal (`.ics`) file to upload.

Calendar * Select an available calendar.

> **Note:** If desired calendar is not listed here, please make sure that you have at least *create* permissions.

Update existing appointments? If checked, all existing appointments in the calendar with same `UniqueID` will be overwritten.



Import Appointments

★ Upload: Browse... No file selected.

★ Calendar:

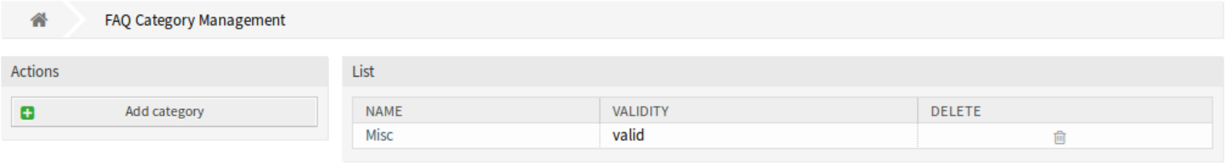
Update existing appointments? □

All existing appointments in the calendar with same UniqueID will be overwritten.

Fig. 5: Import Appointments Screen

10.2 FAQ Category

Use this screen to manage categories available in the knowledge base articles. A fresh OTRS installation already contains a category by default. The category management screen is available in the *FAQ Category* module of the *Administration* group.



NAME	VALIDITY	DELETE
Misc	valid	🗑

Fig. 6: FAQ Category Management Screen

10.2.1 Manage FAQ Categories

To add a category:

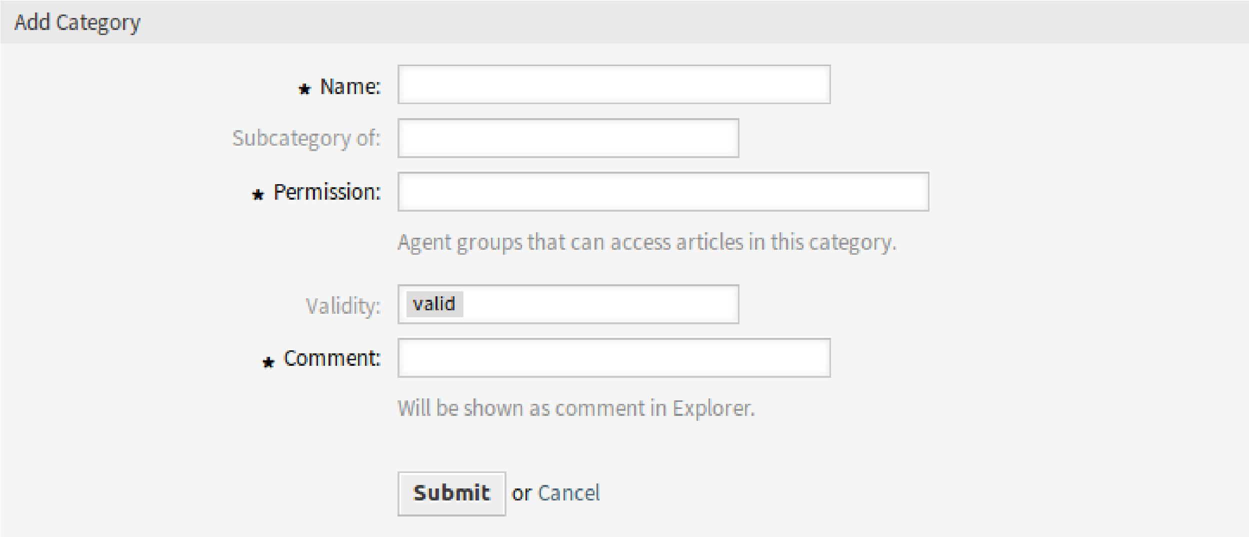
1. Click on the *Add category* button in the left sidebar.
2. Fill in the required fields.
3. Click on the *Submit* button.

To edit a category:

1. Click on a category in the list of categories.
2. Modify the fields.
3. Click on the *Submit* button.

To delete a category:

1. Click on the trash icon in the list of categories.
2. Click on the *Yes* button in the confirmation dialog.



Add Category

★ Name:

Subcategory of:

★ Permission:

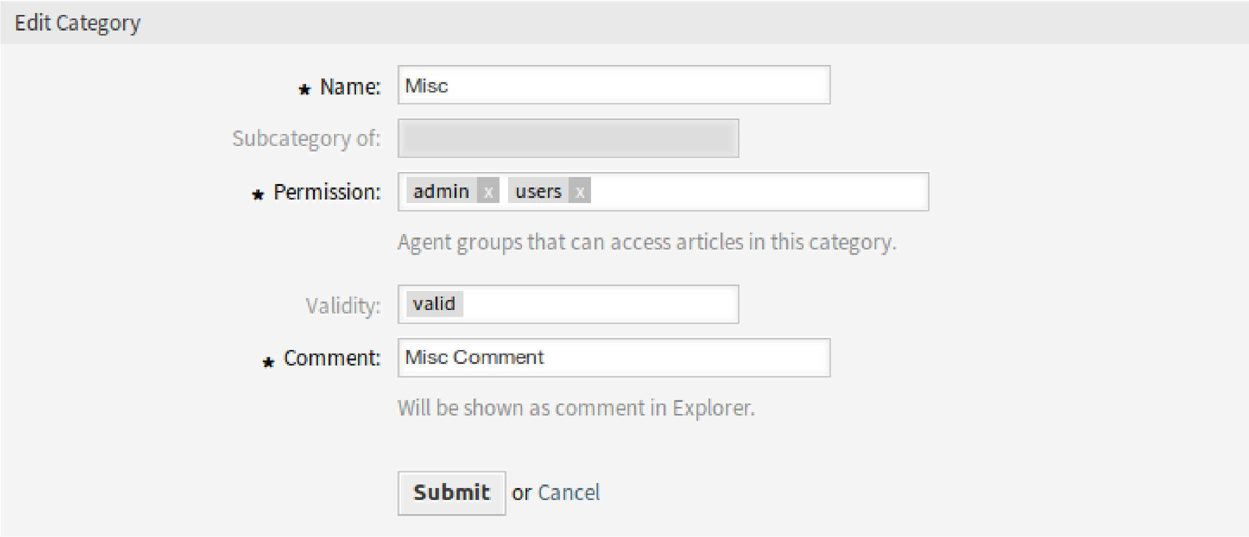
Agent groups that can access articles in this category.

Validity: valid

★ Comment:

Will be shown as comment in Explorer.

Fig. 7: Add Category Screen



Edit Category

★ Name: Misc

Subcategory of:

★ Permission: admin x users x

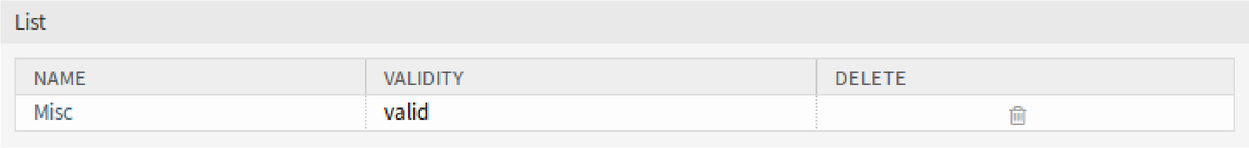
Agent groups that can access articles in this category.

Validity: valid

★ Comment: Misc Comment

Will be shown as comment in Explorer.

Fig. 8: Edit Category Screen



List

NAME	VALIDITY	DELETE
Misc	valid	🗑

Fig. 9: Delete Category Screen

10.2.2 FAQ Category Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Name * The name of this resource. Any type of characters can be entered to this field including uppercase letters and spaces. The name will be displayed in the overview table.

Subcategory of It is possible to add the new category under an existing one as sub-category. This will be displayed as *Parent Category::Child Category*.

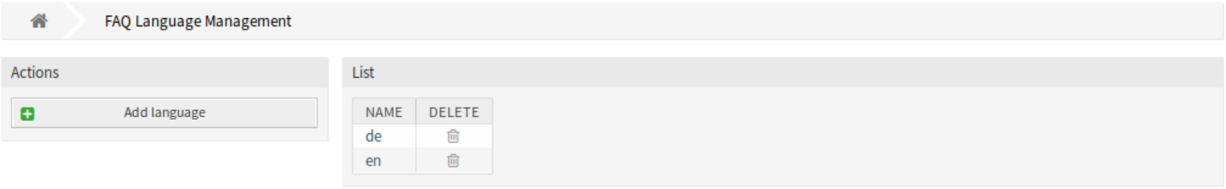
Permission * Agent groups that can access articles in this category.

Validity Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to *valid*. Setting this field to *invalid* or *invalid-temporarily* will disable the use of the resource.

Comment * Add additional information to this resource. It is recommended to always fill this field as a description of the resource with a full sentence for better clarity, because the comment will be also displayed in the overview table.

10.3 FAQ Language

Use this screen to manage languages available in the knowledge base articles. A fresh OTRS installation already contains some languages by default. The language management screen is available in the *FAQ Language* module of the *Administration* group.



NAME	DELETE
de	🗑
en	🗑

Fig. 10: FAQ Language Management Screen

10.3.1 Manage FAQ Languages

To add a language:

1. Click on the *Add language* button in the left sidebar.
2. Fill in the required field.
3. Click on the *Submit* button.

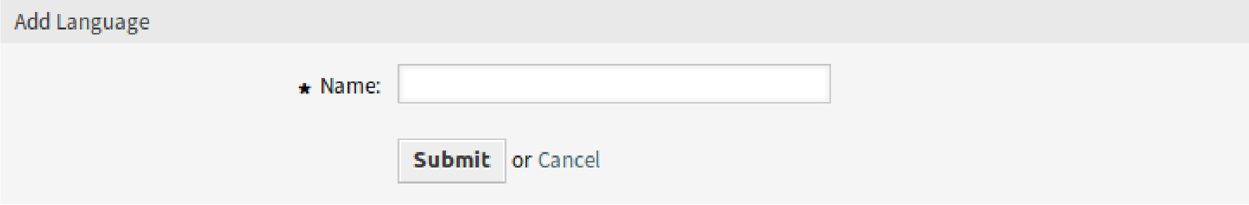
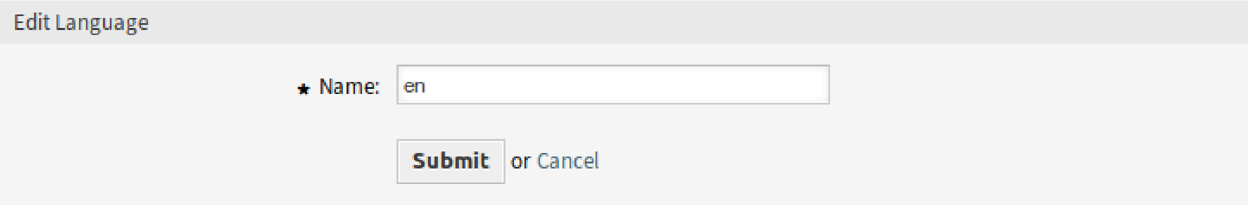


Fig. 11: Add Language Screen

To edit a language:

1. Click on a language in the list of languages.
2. Modify the field.
3. Click on the *Submit* button.



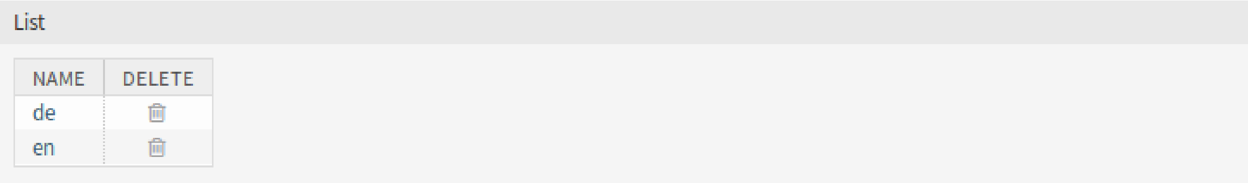
Edit Language

★ Name: en

Fig. 12: Edit Language Screen

To delete a language:

1. Click on the trash icon in the list of languages.
2. Click on the *Yes* button in the confirmation dialog.



List

NAME	DELETE
de	🗑
en	🗑

Fig. 13: Delete Language Screen

10.3.2 FAQ Language Settings

The following setting is available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Name * The ISO 639-1 code of a language.

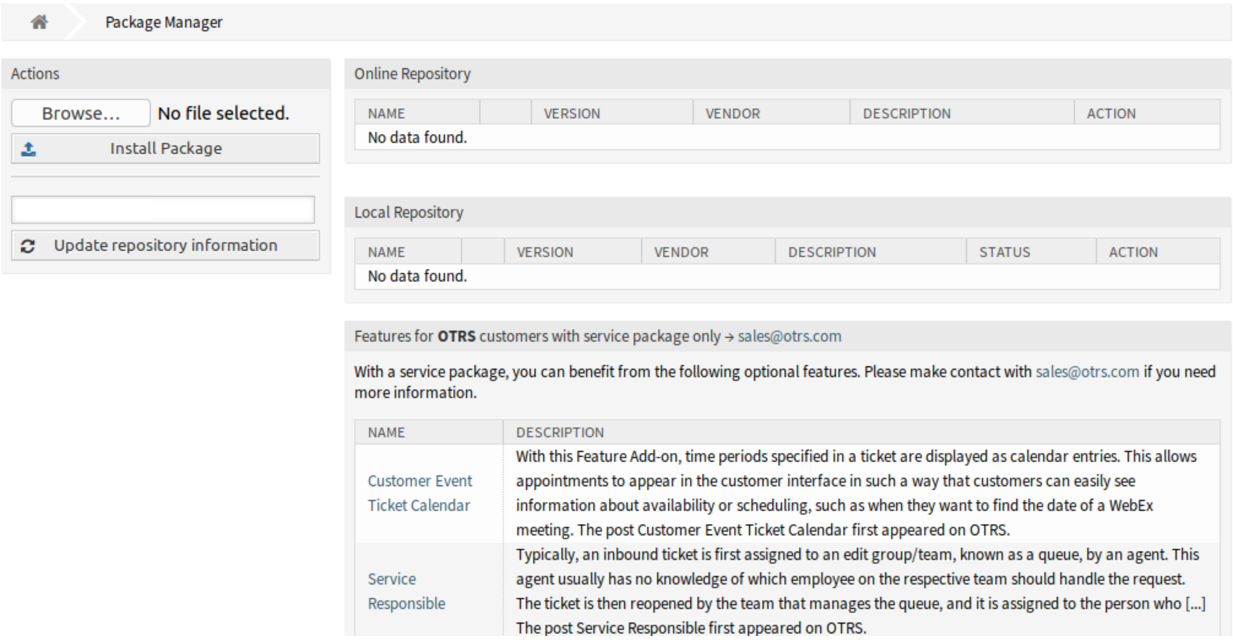
10.4 Package Manager

Modular systems can be extended by adding additional software packages to the framework. Administrators need an easy way to see which features are installed in which version and for sure to add, update and remove packages.

OTRS uses a package manager to perform all package-related activities as mentioned above in the graphical interface.

Note: This module is only available when using *On-Premise* or the *((OTRS)) Community Edition*. To see a list of installed modules, you may also see the *Support Data Collector*.

Use this screen to install and manage packages that extend the functionality of OTRS. The package manager screen is available in the *Package Manager* module of the *Administration* group.



Online Repository

NAME		VERSION	VENDOR	DESCRIPTION	ACTION
No data found.					

Local Repository

NAME		VERSION	VENDOR	DESCRIPTION	STATUS	ACTION
No data found.						

Features for **OTRS** customers with service package only → sales@otrs.com

With a service package, you can benefit from the following optional features. Please make contact with sales@otrs.com if you need more information.

NAME	DESCRIPTION
Customer Event Ticket Calendar	With this Feature Add-on, time periods specified in a ticket are displayed as calendar entries. This allows appointments to appear in the customer interface in such a way that customers can easily see information about availability or scheduling, such as when they want to find the date of a WebEx meeting. The post Customer Event Ticket Calendar first appeared on OTRS.
Service Responsible	Typically, an inbound ticket is first assigned to an edit group/team, known as a queue, by an agent. This agent usually has no knowledge of which employee on the respective team should handle the request. The ticket is then reopened by the team that manages the queue, and it is assigned to the person who [...] The post Service Responsible first appeared on OTRS.

Fig. 14: Package Manager Screen

10.4.1 Manage Packages

Warning: The installation of packages which are not verified by the OTRS Group is not possible by default.

See also:

You can activate the installation of not verified packages in the system configuration setting Package::AllowNotVerifiedPackages.

Install Packages

To install a package from online repository:

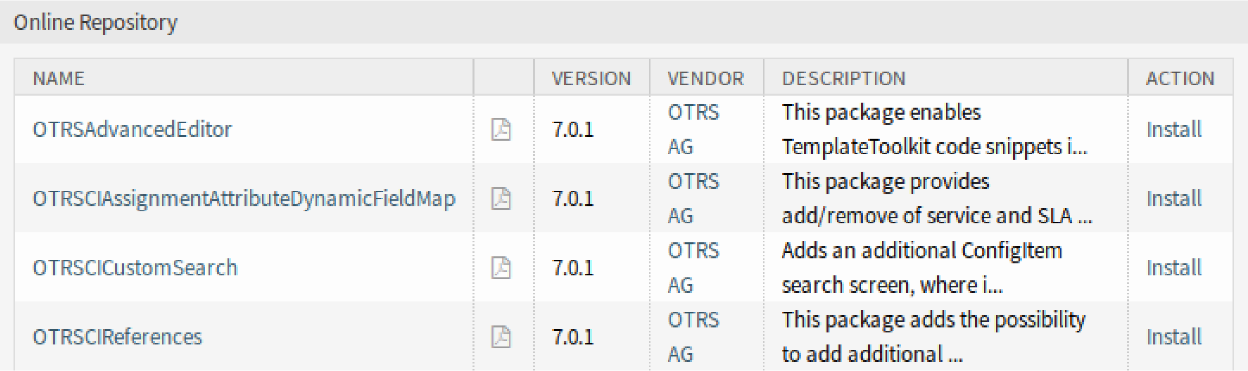
1. Select an online repository from the drop-down in the left sidebar.
2. Click on the *Update repository information* button to refresh the available package list.
3. Select a package from the *Online Repository* widget and click on the *Install* in the last column.
4. Follow the installation instructions.
5. After installation, the package is displayed in the *Local Repository* widget.

See also:

The repository list can be changed in system configuration setting Package::RepositoryList.

To install a package from file:

1. Click on the *Browse*⋯ button in the left sidebar.
2. Select an `.opm` file from your local file system.

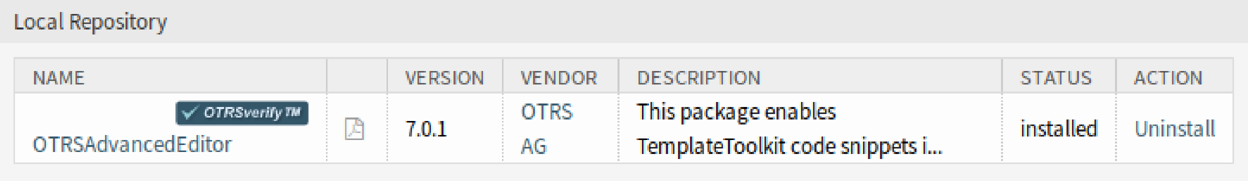


Online Repository

NAME		VERSION	VENDOR	DESCRIPTION	ACTION
OTRSAdvancedEditor		7.0.1	OTRS AG	This package enables TemplateToolkit code snippets i...	Install
OTRSCIAssignmentAttributeDynamicFieldMap		7.0.1	OTRS AG	This package provides add/remove of service and SLA ...	Install
OTRSCICustomSearch		7.0.1	OTRS AG	Adds an additional ConfigItem search screen, where i...	Install
OTRSCIReferences		7.0.1	OTRS AG	This package adds the possibility to add additional ...	Install

Fig. 15: Online Repository Widget

3. Click on the *Install Package* button.
4. Follow the installation instructions.
5. After installation, the package is displayed in the *Local Repository* widget.



Local Repository

NAME		VERSION	VENDOR	DESCRIPTION	STATUS	ACTION
✓ OTRSverify™ OTRSAdvancedEditor		7.0.1	OTRS AG	This package enables TemplateToolkit code snippets i...	installed	Uninstall

Fig. 16: Local Repository Widget

Update Packages

To update a package from online repository:

1. Check the available packages in the *Online Repository* widget if there is *Update* in the *Action* column.
2. Click on the *Update* link.
3. Follow the update instructions.
4. After updating, the package is displayed in the *Local Repository* widget.

To update a package from file:

1. Click on the *Browse*··· button in the left sidebar.
2. Select an `.opm` file which is newer than the installed package.
3. Click on the *Install Package* button.
4. Follow the update instructions.
5. After updating, the package is displayed in the *Local Repository* widget.

To update all packages:

1. Click on the *Update all installed packages* button in the left sidebar.
2. Follow the update instructions.
3. After updating, the package is displayed in the *Local Repository* widget.

This feature reads the information of all defined package repositories and determines if there is a new version for every installed package in the system and calculates the correct order to update the packages respecting all other package dependencies, even if new versions of existing packages require new packages not yet installed in the system.

Note: If there are packages installed that do not have a corresponding repository defined in the system, they can not be updated by this feature and will be marked as failed (due to the missing on-line repository).

Reinstall Packages

If at least one of the package files are modified locally, the package manager marks the package as broken, and need to reinstall.

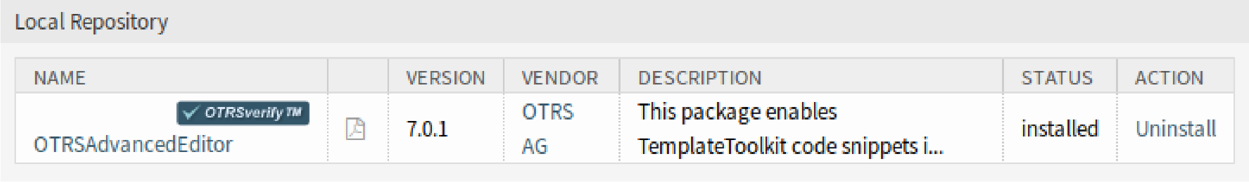
To reinstall a package:

1. Select the package from the *Local Repository* widget that are marked for reinstall.
2. Click on the *Reinstall* link in the *Action* column.
3. Follow the installation instructions.

Uninstall Packages

To uninstall a package:

1. Select the package from the *Local Repository* widget.
2. Click on the *Uninstall* link in the *Action* column.
3. Follow the uninstall instructions.



Local Repository

NAME		VERSION	VENDOR	DESCRIPTION	STATUS	ACTION
✓ OTRSverify™ OTRSAdvancedEditor		7.0.1	OTRS AG	This package enables TemplateToolkit code snippets i...	installed	Uninstall

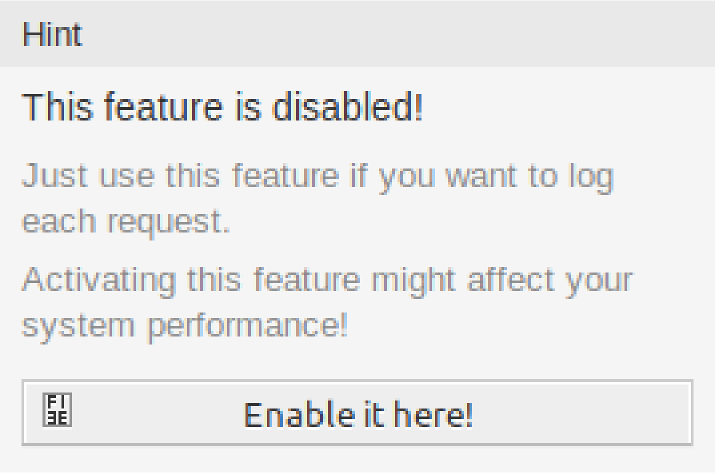
Fig. 17: Local Repository Widget

10.5 Performance Log

Performance is always a crucial topic for web-based applications. Administrators need to have the possibility to see which activities use which time to execute over time to identify potential increases over time and take proper measures.

OTRS supports this requirement with the performance log. The performance log can, when it is activated, log activities and display various activity types and their min/max/average response time and a number of requests for different time frames.

Use this screen to view the performance log of OTRS. The log overview screen is available in the *Performance Log* module of the *Administration* group.



Hint

This feature is disabled!

Just use this feature if you want to log each request.

Activating this feature might affect your system performance!

Fig. 18: Enable Performance Log Support

Note: To be able to see performance log in OTRS, you have to enabled its setting first.

If the performance log is enabled, OTRS collects all the information about requests and responses in an overview table.

Clicking on an entry will show the details.

10.6 Session Management

Administrators of web-based applications need to have access to the information who's connected to the system and, if required, delete unwanted sessions.

OTRS offers the session management to quickly get an overview of agent and customer sessions, unique agents and customers currently logged in and the ability to kill sessions with just a mouse click.

Use this screen to manage logged in user sessions in the system. The session management screen is available in the *Session Management* module of the *Administration* group.

The first widget lists sessions and tokens. Sessions are used for logins to the administrator interface, while logins to the agent interface, to the external interface or to the generic interface use tokens.

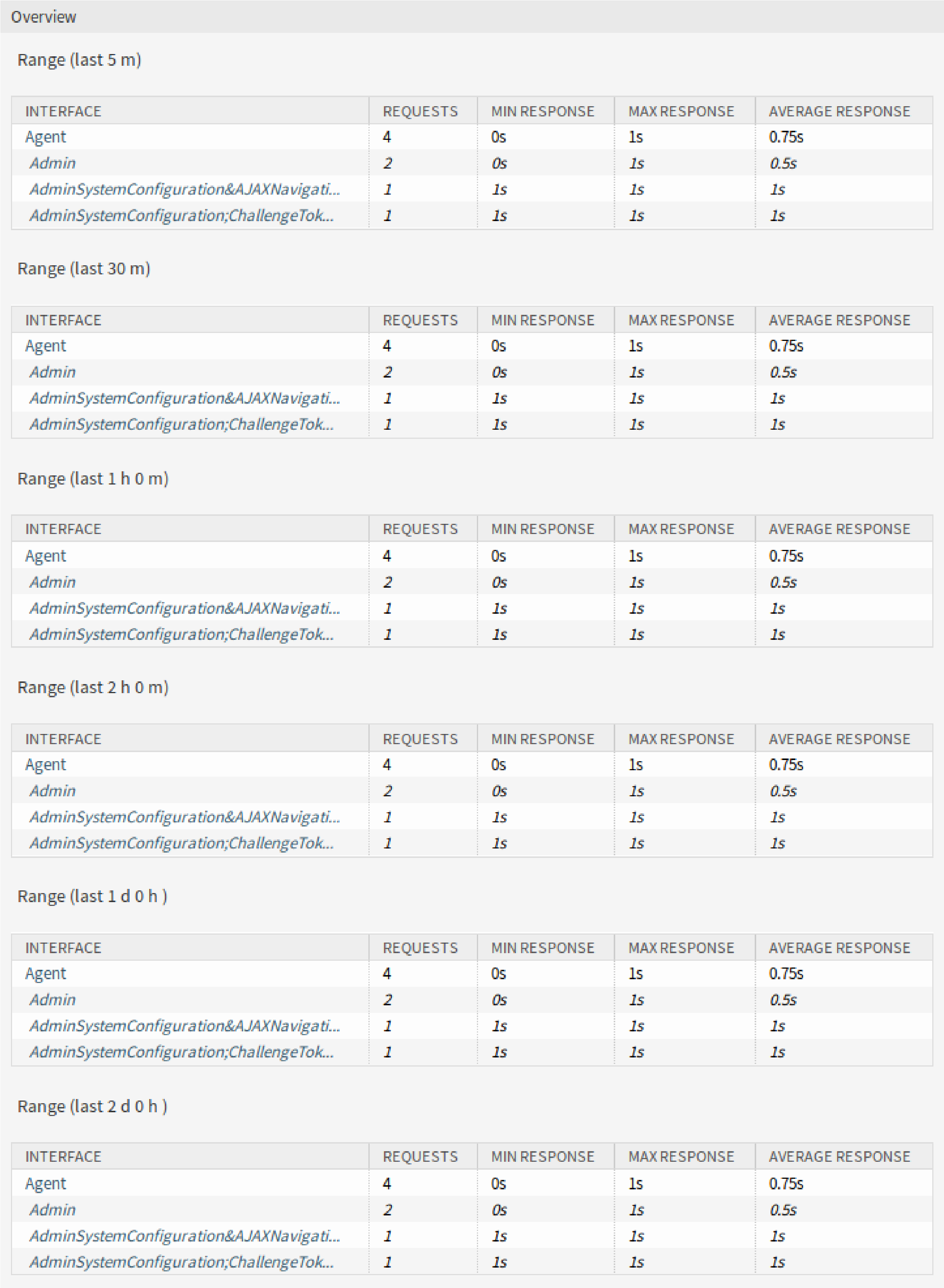
The widget *List of tokens (one time only)* shows special one time tokens used for the password recovery and two-factor setup. These tokens are short-lived and can be used for only a single action.

10.6.1 Manage Sessions

To see a logged in user session:

1. Select a logged in user from the list of sessions.
2. Click on the token.
3. See the details.

To kill a session:



Overview

Range (last 5 m)

INTERFACE	REQUESTS	MIN RESPONSE	MAX RESPONSE	AVERAGE RESPONSE
Agent	4	0s	1s	0.75s
Admin	*2*	*0s*	*1s*	*0.5s*
AdminSystemConfiguration&AJAXNavigati...	*1*	*1s*	*1s*	*1s*
AdminSystemConfiguration;ChallengeTok...	*1*	*1s*	*1s*	*1s*

Range (last 30 m)

INTERFACE	REQUESTS	MIN RESPONSE	MAX RESPONSE	AVERAGE RESPONSE
Agent	4	0s	1s	0.75s
Admin	*2*	*0s*	*1s*	*0.5s*
AdminSystemConfiguration&AJAXNavigati...	*1*	*1s*	*1s*	*1s*
AdminSystemConfiguration;ChallengeTok...	*1*	*1s*	*1s*	*1s*

Range (last 1 h 0 m)

INTERFACE	REQUESTS	MIN RESPONSE	MAX RESPONSE	AVERAGE RESPONSE
Agent	4	0s	1s	0.75s
Admin	*2*	*0s*	*1s*	*0.5s*
AdminSystemConfiguration&AJAXNavigati...	*1*	*1s*	*1s*	*1s*
AdminSystemConfiguration;ChallengeTok...	*1*	*1s*	*1s*	*1s*

Range (last 2 h 0 m)

INTERFACE	REQUESTS	MIN RESPONSE	MAX RESPONSE	AVERAGE RESPONSE
Agent	4	0s	1s	0.75s
Admin	*2*	*0s*	*1s*	*0.5s*
AdminSystemConfiguration&AJAXNavigati...	*1*	*1s*	*1s*	*1s*
AdminSystemConfiguration;ChallengeTok...	*1*	*1s*	*1s*	*1s*

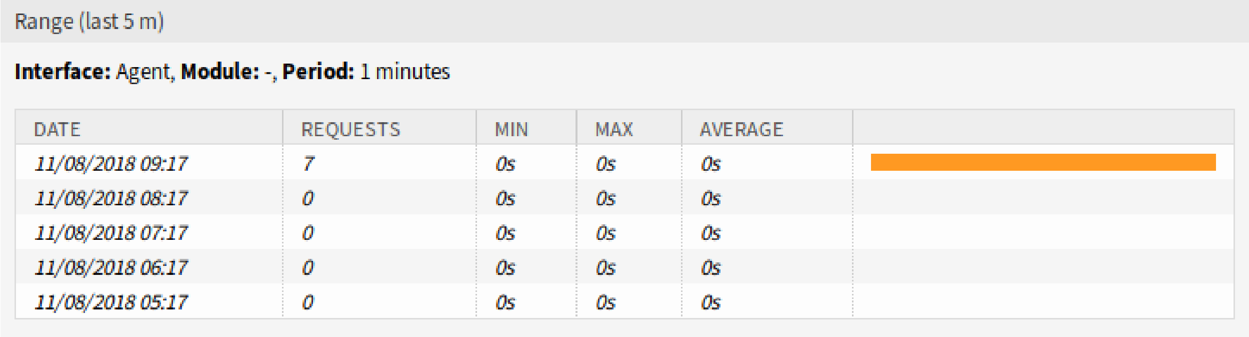
Range (last 1 d 0 h)

INTERFACE	REQUESTS	MIN RESPONSE	MAX RESPONSE	AVERAGE RESPONSE
Agent	4	0s	1s	0.75s
Admin	*2*	*0s*	*1s*	*0.5s*
AdminSystemConfiguration&AJAXNavigati...	*1*	*1s*	*1s*	*1s*
AdminSystemConfiguration;ChallengeTok...	*1*	*1s*	*1s*	*1s*

Range (last 2 d 0 h)

INTERFACE	REQUESTS	MIN RESPONSE	MAX RESPONSE	AVERAGE RESPONSE
Agent	4	0s	1s	0.75s
Admin	*2*	*0s*	*1s*	*0.5s*
AdminSystemConfiguration&AJAXNavigati...	*1*	*1s*	*1s*	*1s*
AdminSystemConfiguration;ChallengeTok...	*1*	*1s*	*1s*	*1s*

Fig. 19: Performance Log Screen

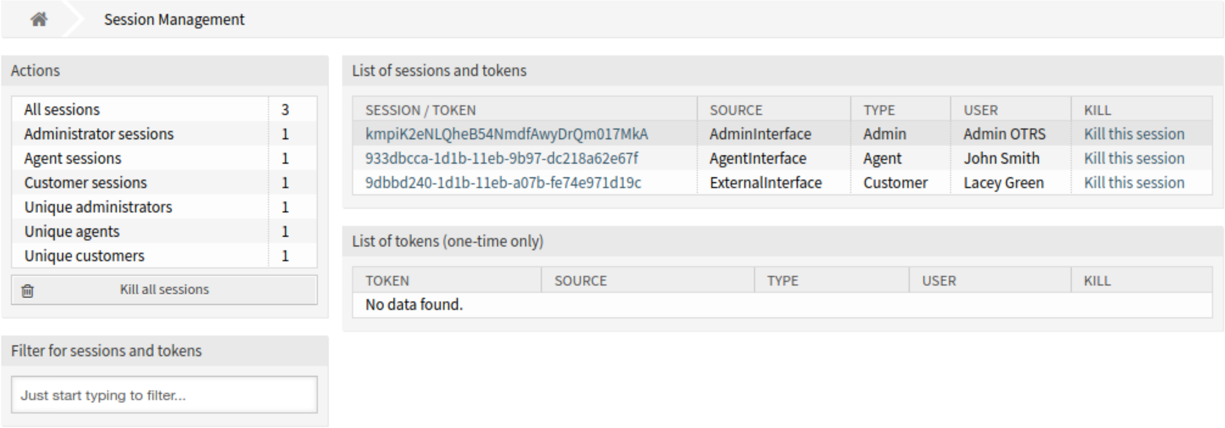


Range (last 5 m)

Interface: Agent, **Module:** -, **Period:** 1 minutes

DATE	REQUESTS	MIN	MAX	AVERAGE	
11/08/2018 09:17	*7*	*0s*	*0s*	*0s*	
11/08/2018 08:17	*0*	*0s*	*0s*	*0s*	
11/08/2018 07:17	*0*	*0s*	*0s*	*0s*	
11/08/2018 06:17	*0*	*0s*	*0s*	*0s*	
11/08/2018 05:17	*0*	*0s*	*0s*	*0s*	

Fig. 20: Performance Log Details Screen



Session Management

Actions

All sessions	3
Administrator sessions	1
Agent sessions	1
Customer sessions	1
Unique administrators	1
Unique agents	1
Unique customers	1

Filter for sessions and tokens

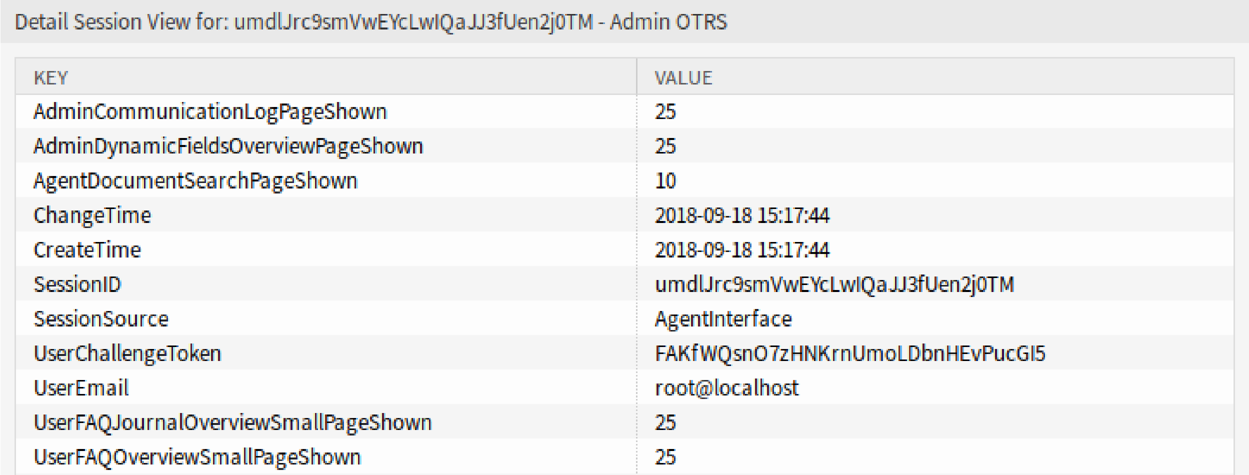
List of sessions and tokens

SESSION / TOKEN	SOURCE	TYPE	USER	KILL
kmpiK2eNLQheB54NmdfAwyDrQm017MkA	AdminInterface	Admin	Admin OTRS	Kill this session
933dbcca-1d1b-11eb-9b97-dc218a62e67f	AgentInterface	Agent	John Smith	Kill this session
9dbbd240-1d1b-11eb-a07b-fe74e971d19c	ExternalInterface	Customer	Lacey Green	Kill this session

List of tokens (one-time only)

TOKEN	SOURCE	TYPE	USER	KILL
No data found.				

Fig. 21: Session Management Screen

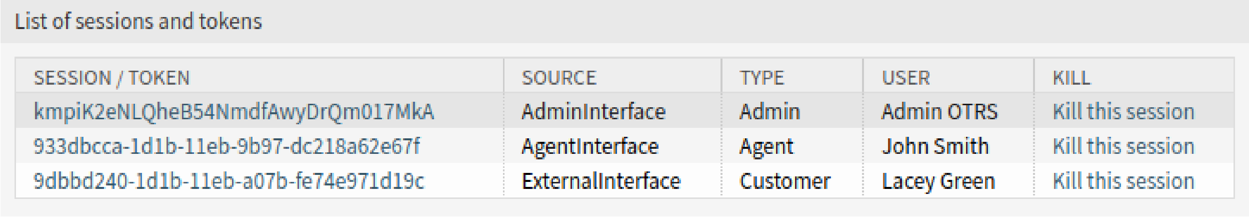


Detail Session View for: umdlJrc9smVwEYcLwlQaJJ3fUen2j0TM - Admin OTRS

KEY	VALUE
AdminCommunicationLogPageShown	25
AdminDynamicFieldsOverviewPageShown	25
AgentDocumentSearchPageShown	10
ChangeTime	2018-09-18 15:17:44
CreateTime	2018-09-18 15:17:44
SessionID	umdlJrc9smVwEYcLwlQaJJ3fUen2j0TM
SessionSource	AgentInterface
UserChallengeToken	FAKfWQsnO7zHNKrnUmoLDbnHEvPucGI5
UserEmail	root@localhost
UserFAQJournalOverviewSmallPageShown	25
UserFAQOverviewSmallPageShown	25

Fig. 22: Session Management Details Screen

1. Select a logged in user from the list of sessions.
2. Click on the *Kill this session* link in the *Kill* column.



List of sessions and tokens

SESSION / TOKEN	SOURCE	TYPE	USER	KILL
kmpiK2eNLQheB54NmdfAwyDrQm017MkA	AdminInterface	Admin	Admin OTRS	Kill this session
933dbcca-1d1b-11eb-9b97-dc218a62e67f	AgentInterface	Agent	John Smith	Kill this session
9dbbd240-1d1b-11eb-a07b-fe74e971d19c	ExternalInterface	Customer	Lacey Green	Kill this session

Fig. 23: Session Kill Screen

Note: The current administrator session is displayed with a darker gray row background. This allows you to identify your own session in the list easier.

Warning: Clicking the *Kill this session* link removes the session immediately without confirmation. The unsaved work of the user will be lost!

To kill all sessions:

1. Click on the *Kill all sessions* button in the left sidebar.

Warning: Clicking the *Kill all session* link removes all sessions or tokens immediately without confirmation. The unsaved work of the users will be lost!

Note: If several users are logged in to the system, use the filter box to find a particular session by just typing the name to filter.

10.7 SQL Box

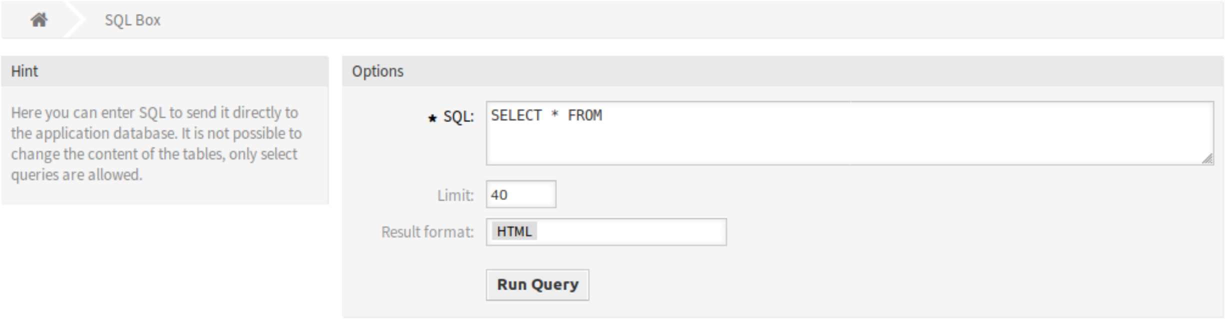
Note: This feature is not enabled by default. Activate the system configuration setting Frontend::Module###AdminSelectBox to enable this feature.

In a ticket system, it is usually possible to have statistics that show a summarized view of ticket information when needed. Sometimes, it is however required to access the database directly to have even more individual reports, allow external statistic tools to query information from the system or perform in-depth analysis of a ticket behavior.

Direct access to the database requires access to the command line which an administrator may not have. In addition to the username and password for the command line access, which is not given by all organizations, the username and password for the database are needed. These hurdles can prevent an administrator from using the database for more complex searches and operations.

OTRS offers application administrators the SQL Box in the graphical interface. It allows read access to the database. All results can be seen in the GUI or exported to CSV/Excel files.

Use this screen to query SQL statements in the system. The SQL box screen is available in the *SQL Box* module of the *Administration* group.



Hint

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

Options

★ SQL: SELECT * FROM

Limit: 40

Result format: HTML

Fig. 24: SQL Box Screen

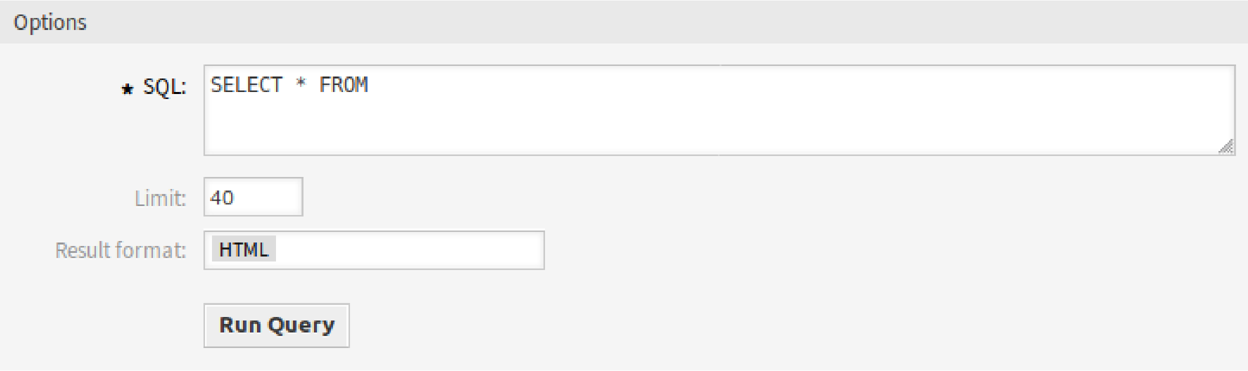
10.7.1 Query SQL statements

Note: The SQL statements entered here are sent directly to the application database. By default, it is not possible to change the content of the tables, only SELECT queries are allowed.

Warning: It is possible to modify the application database via SQL box. To do this, you have to enabled the system configuration setting AdminSelectBox::AllowDatabaseModification. Activate it to your own risk!

To execute an SQL statement:

1. Enter the SQL statement into the SQL box.
2. Select the result format.
3. Click on the *Run Query* button.



Options

★ SQL: SELECT * FROM

Limit: 40

Result format: HTML

Fig. 25: SQL Box Widget

10.7.2 SQL Settings

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

SQL * The SQL statement to be queried.

Limit Enter a number to limit the result to at most this number of rows. Leaving this field empty means there is no limit.

> **Note:** Don’t use `LIMIT` inside the SQL statement. Always use this field to limit the number of results.

Result format The format of the SQL statement result.

HTML The results are visible below the SQL box in a new widget.

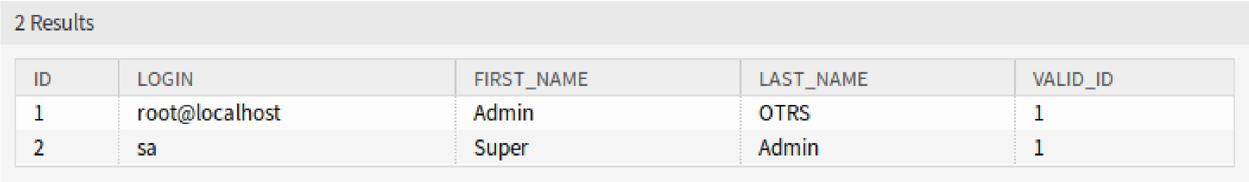
CSV The results can be downloaded in comma separated plain text format.

Excel The results can be downloaded as Microsoft Excel spreadsheet.

10.7.3 SQL Examples

To list some information about agents and output the results as HTML:

```sql
SELECT id, login , first_name, last_name, valid_id FROM users
```



2 Results

ID	LOGIN	FIRST_NAME	LAST_NAME	VALID_ID
1	root@localhost	Admin	OTRS	1
2	sa	Super	Admin	1

Fig. 26: SQL Box Result

To list all tables, you need to leave empty the *Limit* field and run the following query:

```sql
SHOW TABLES
```

To show the structure of the `users` table, you need to limit the results to 1 and run the following query (see the table header for the columns):

```sql
SELECT * FROM users
```

10.8 System Configuration

Modern systems have many ways to configure their behavior. Some use configuration files edited on the command line, and some use a graphical interface (and save the information to configuration files in the background), yet others use a database. Maintaining changes and auditing can sometimes be an issue, as it is not always clear who made a change. Making bulk changes is not always possible, and rolling back changes a chore.

OTRS uses a comfortable graphical interface to configure the system. All changes to the default system configuration are stored in the database and can be audited (who changed a setting and when, what was the old and what is the new value) and rolled back to a previous state in case of misconfiguration.

Comfortable search allows finding the needed settings quickly and easily.

See also:

By using the System Configuration History package, you can easily roll back changes made by users. Contact sales@otrs.com to add this feature to your system.

Use this screen to manage the system configuration settings. OTRS brings about 1750 configuration settings. The system configuration management screen is available in the *System Configuration* module of the *Administration* group.

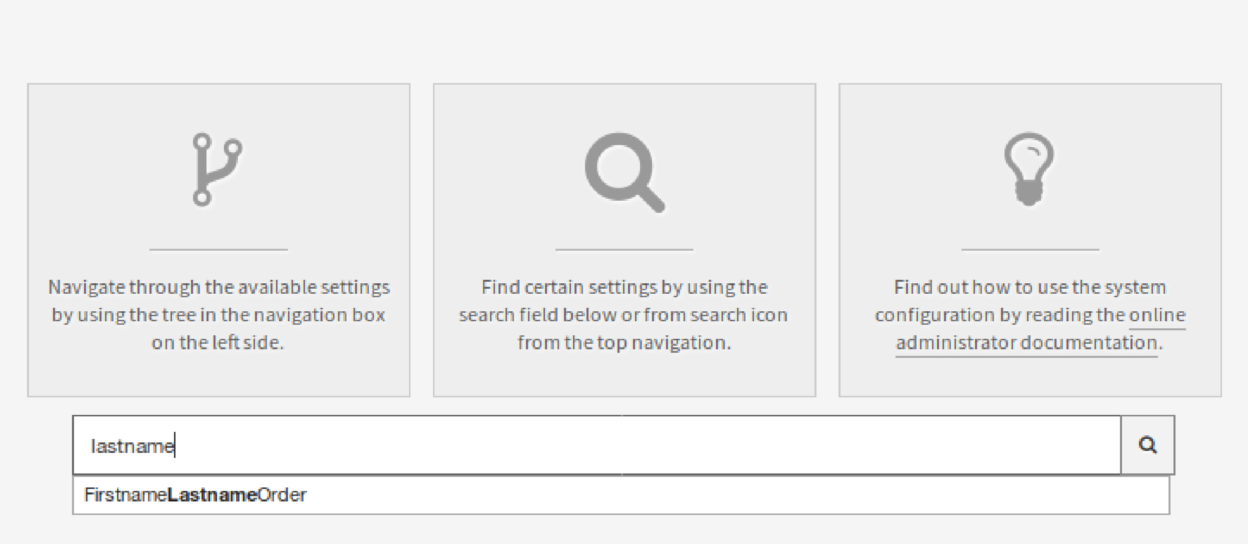
10.8.1 Manage System Configurations

Note: For security reasons, the configuration settings for the database connection cannot be changed in the graphical user interface of the system configuration. These have to be set manually in `Kernel/Config.pm`.

To modify a system configuration, you need to do several steps. The following example shows you, how to find, modify, deploy and reset the system configuration `FirstnameLastnameOrder`.

1. Find the system configuration by entering a search term `lastname` into to search box.

   With the full-text search, all configuration settings can be scanned for one or more keywords. The full-text search not only searches through the names of the configuration settings, but also the descriptions and values. This allows an element to be found easily even if its name is unknown.



Navigate through the available settings by using the tree in the navigation box on the left side.

Find certain settings by using the search field below or from search icon from the top navigation.

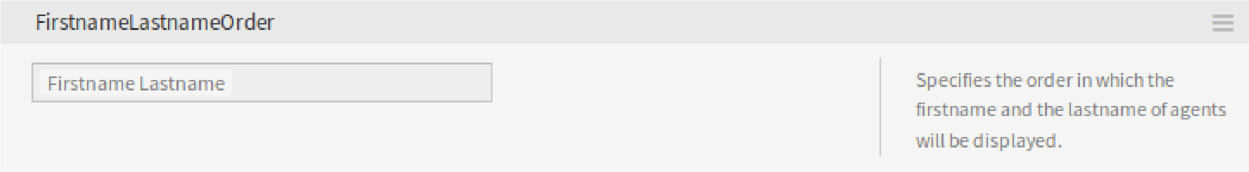
Find out how to use the system configuration by reading the online administrator documentation.

lastname

FirstnameLastnameOrder

Fig. 27: System Configuration - Search For Setting

2. Select the setting from the search results.
3. Click on the header of the widget to see the options.
4. Hover the mouse over the widget body to see the *Edit this setting* button.

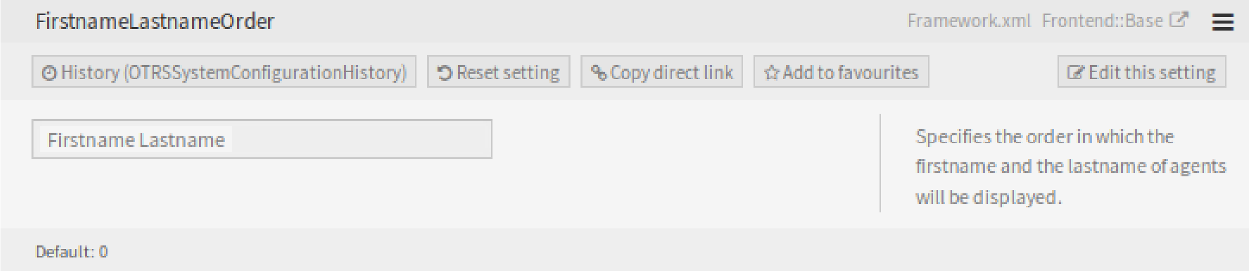


FirstnameLastnameOrder

Firstname Lastname

Specifies the order in which the firstname and the lastname of agents will be displayed.

Fig. 28: System Configuration - Setting Found



FirstnameLastnameOrder

Firstname Lastname

Specifies the order in which the firstname and the lastname of agents will be displayed.

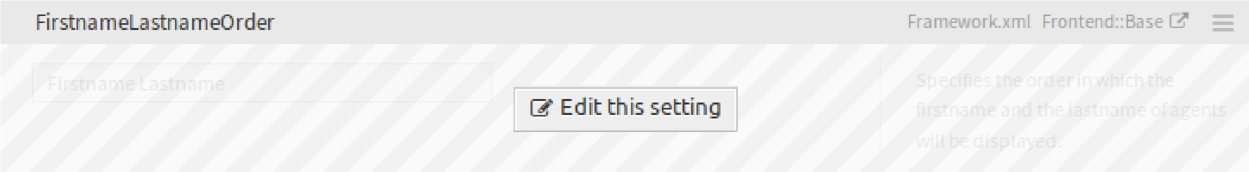
Default: 0

Fig. 29: System Configuration - Setting Expanded

5. Click on the *Edit this setting* button to activate the edit mode. In edit mode the widget gets an orange border on the left.

   Note: If a setting is currently edited by another administrator, it is not possible to get access to the edit mode for that setting until the other administrator finished their work.

6. Change the value of the setting. Editing can be cancelled by clicking the *Cancel* button on the right or hitting the *Escape* key on your keyboard. When editing is cancelled, all changes made during the current editing session are discarded.
7. Click on the *Save* button. If the modification is saved, the widget gets a green border on the left.
8. Go back and click on the *Deployment* button in the left sidebar. You are also notified in the notification bar, that you have undeployed settings.
9. Review the changes.
10. You can click on the   button in the top right corner to compare the changes side-by-side.
11. Click on the *Deploy selected changes* button. If several settings are changed, it is possible to deploy only the selected settings.
12. Add a deployment comment, that explain for other administrators, what is changed and why. Use full sentence here.

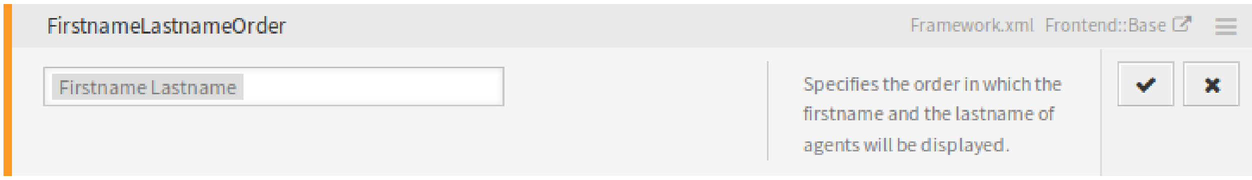


FirstnameLastnameOrder

Firstname Lastname

Specifies the order in which the firstname and the lastname of agents will be displayed.

Fig. 30: System Configuration - Setting Hovered

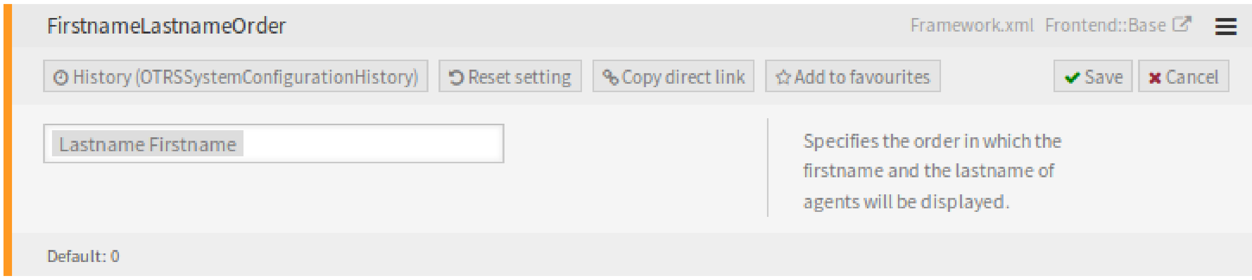

FirstnameLastnameOrder

Framework.xml Frontend::Base

Firstname Lastname

Specifies the order in which the firstname and the lastname of agents will be displayed.

Fig. 31: System Configuration - Setting Clicked


FirstnameLastnameOrder

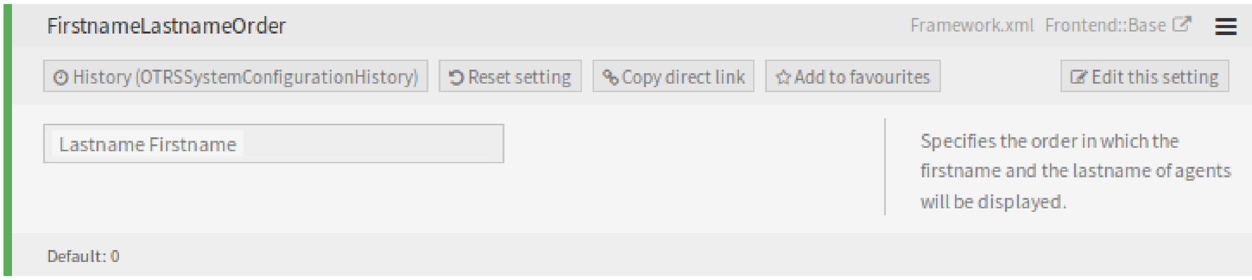
Framework.xml Frontend::Base

Lastname Firstname

Specifies the order in which the firstname and the lastname of agents will be displayed.

Default: 0

Fig. 32: System Configuration - Setting Modified


FirstnameLastnameOrder

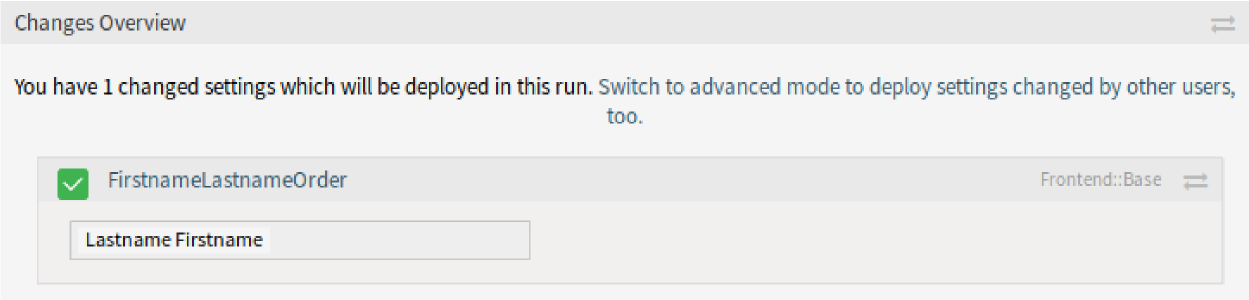
Framework.xml Frontend::Base

Lastname Firstname

Specifies the order in which the firstname and the lastname of agents will be displayed.

Default: 0

Fig. 33: System Configuration - Setting Saved


Changes Overview

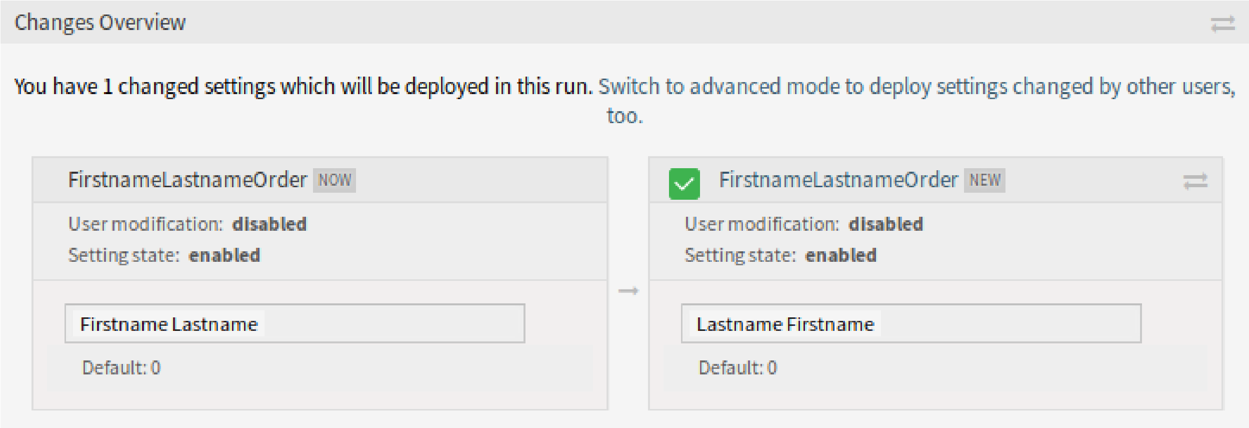
You have 1 changed settings which will be deployed in this run. Switch to advanced mode to deploy settings changed by other users, too.

☑ FirstnameLastnameOrder

Frontend::Base

Lastname Firstname

Fig. 34: System Configuration - Setting Changes



Changes Overview

You have 1 changed settings which will be deployed in this run. Switch to advanced mode to deploy settings changed by other users, too.

FirstnameLastnameOrder NOW

User modification: **disabled**
Setting state: **enabled**

Firstname Lastname

Default: 0

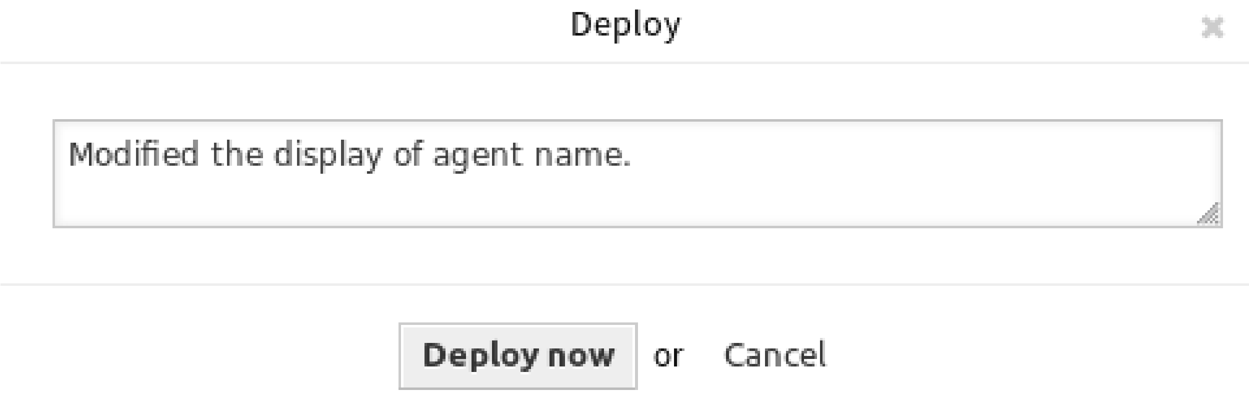
☑ FirstnameLastnameOrder NEW

User modification: **disabled**
Setting state: **enabled**

Lastname Firstname

Default: 0

Fig. 35: System Configuration - Setting Different

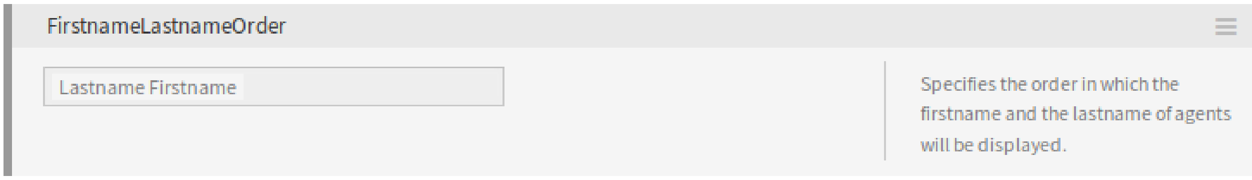


Deploy

Modified the display of agent name.

Fig. 36: System Configuration - Deploy Setting

13. Go back and search again the term `lastname` to find the modified setting. The widget has a gray border on the left to indicate, this setting is modified.



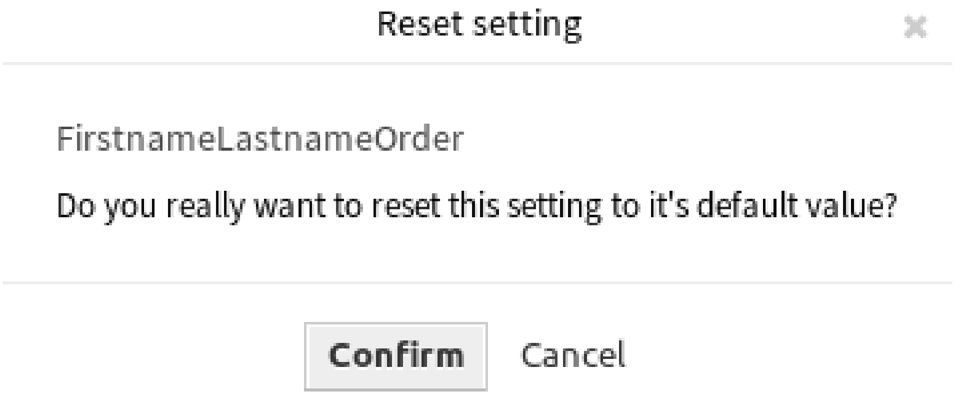
FirstnameLastnameOrder

Lastname Firstname

Specifies the order in which the firstname and the lastname of agents will be displayed.

Fig. 37: System Configuration - Setting Deployed

14. To reset the setting, click on the header of the widget to see the options. Then click on the *Reset setting* button.



Reset setting

FirstnameLastnameOrder

Do you really want to reset this setting to it's default value?

Fig. 38: System Configuration - Reset Setting

15. Click on the *Confirm* button.
16. Deploy the settings.

10.8.2 Using The Navigation Tree

Each configuration setting is classified by a category and a navigation group. Navigation groups are individual elements in the main navigation tree. By selecting one of these navigation entries, all settings assigned to the selected group will be shown. As long as no extensions are installed, the category selection is automatically hidden, but as soon as a package is installed which brings its own configuration settings (such as ITSM modules or Survey), the category selection will be revealed. Selecting a category makes the main navigation tree show only the navigation groups belonging to the selected category.

To expand an element, click on the arrow next to it. The number between the parentheses indicates how many settings belongs to this element. If an element has no number, this element is only a wrapper category. It doesn't have settings, it has only sub-categories.

Using the navigation tree results the same as search for a setting. If you would like to see for a setting to which group belongs to, expand it by clicking on the header of the widget.

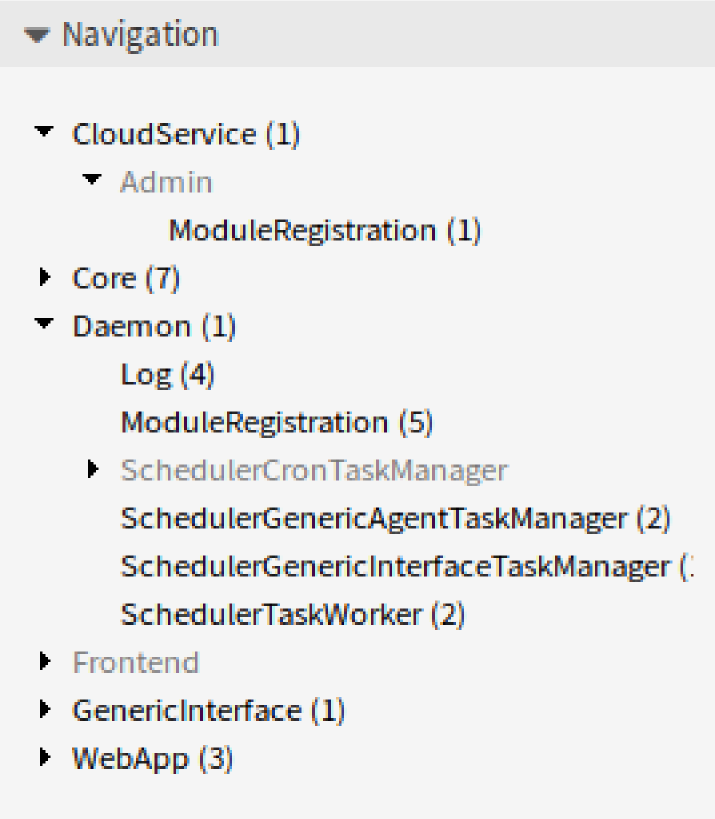
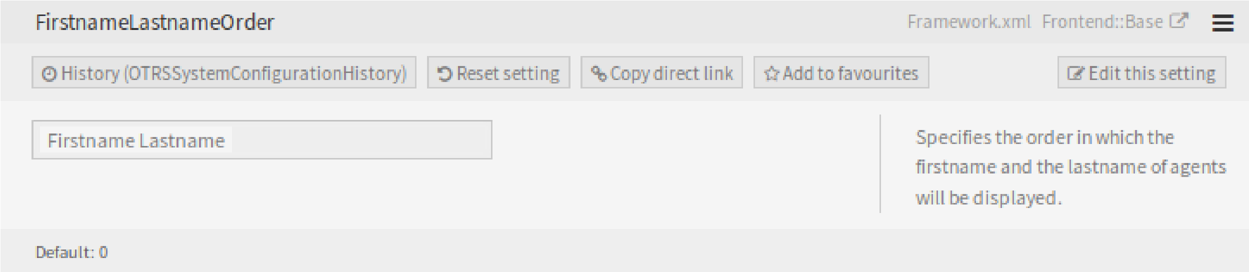


Fig. 39: System Configuration Navigation Tree



FirstnameLastnameOrder

Firstname Lastname

Specifies the order in which the firstname and the lastname of agents will be displayed.

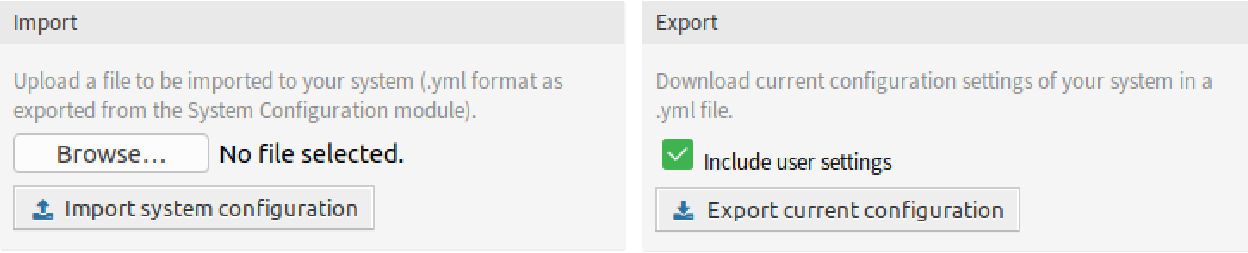
Default: 0

Fig. 40: System Configuration - Setting Expanded

For example `FirstnameLastnameOrder` can be found in *Frontend* → *Base*.

10.8.3 Import And Export System Configurations

Click on the *Import & Export* button in the left sidebar to access the import-export screen.



Import

Upload a file to be imported to your system (.yml format as exported from the System Configuration module).

No file selected.

Export

Download current configuration settings of your system in a .yml file.

☑ Include user settings

Fig. 41: System Configuration - Import and Export

To export the system configurations:

1. Click on the *Export current configuration* button in the *Export* widget.
2. Save the `Export_Current_System_Configuration.yml` file to your local file system.
3. Rename the file to a more descriptive name.

To import the system configurations:

1. Click on the *Browse…* button in the *Import* widget.
2. Select a previously exported `.yml` file.
3. Click on the *Import system configuration* button.

10.8.4 Business Object Configuration

The following section contains a description of the business object configuration, including the business object lists, business object detail views, business cards and forms.

Business Object Lists

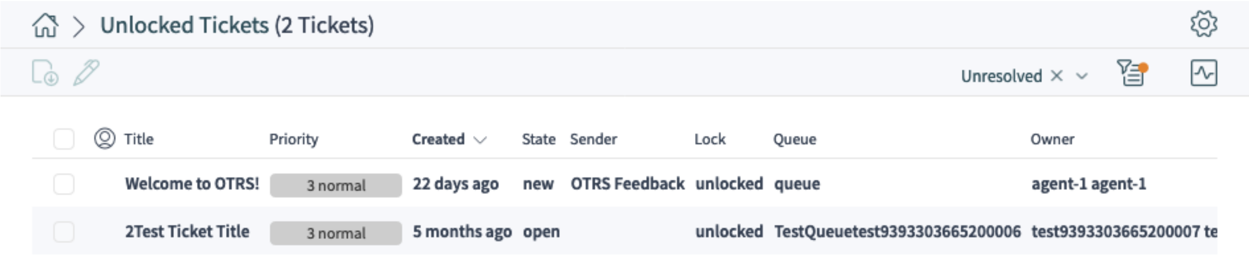
The business object lists provide a tabular view of items with support for configurable columns, sorting and filtering. These lists can be used in many contexts, including as stand-alone screens, widgets, actions, etc.

The default configuration of business object lists can be defined in several places, depending on the screen or element where they are used.

AgentFrontend::<BusinessObjectListType>::<SlugName>###DefaultConfig Standalone or static list screens which are usable via their own URL.

The following system configuration settings are relevant:

- AgentFrontend::KnowledgeBaseArticleList::Static###DefaultConfig
- AgentFrontend::KnowledgeBaseArticleList::Added###DefaultConfig
- AgentFrontend::KnowledgeBaseArticleList::Updated###DefaultConfig
- AgentFrontend::KnowledgeBaseArticleList::Rated###DefaultConfig



Unlocked Tickets (2 Tickets)

□	Title	Priority	Created	State	Sender	Lock	Queue	Owner
□	Welcome to OTRS!	3 normal	22 days ago	new	OTRS Feedback	unlocked	queue	agent-1 agent-1
□	2Test Ticket Title	3 normal	5 months ago	open		unlocked	TestQueuetest9393303665200006	test9393303665200007 te

Fig. 42: An Example of the *Unlocked Tickets* Screen

- AgentFrontend::KnowledgeBaseArticleList::Custom1###DefaultConfig
- AgentFrontend::KnowledgeBaseArticleList::Custom2###DefaultConfig
- AgentFrontend::KnowledgeBaseArticleList::Custom3###DefaultConfig
- AgentFrontend::KnowledgeBaseArticleList::Custom4###DefaultConfig
- AgentFrontend::KnowledgeBaseArticleList::Custom5###DefaultConfig
- AgentFrontend::TicketList::Static###DefaultConfig
- AgentFrontend::TicketList::Unresolved###DefaultConfig
- AgentFrontend::TicketList::Unlocked###DefaultConfig
- AgentFrontend::TicketList::Reminders###DefaultConfig
- AgentFrontend::TicketList::Escalations###DefaultConfig
- AgentFrontend::TicketList::Created###DefaultConfig
- AgentFrontend::TicketList::Closed###DefaultConfig
- AgentFrontend::TicketList::Queues###DefaultConfig
- AgentFrontend::TicketList::Legacyservice###DefaultConfig
- AgentFrontend::TicketList::Legacystatus###DefaultConfig
- AgentFrontend::TicketList::Legacyescalations###DefaultConfig
- AgentFrontend::TicketList::Legacyresponsible###DefaultConfig
- AgentFrontend::TicketList::Legacywatcher###DefaultConfig
- AgentFrontend::TicketList::Custom1###DefaultConfig
- AgentFrontend::TicketList::Custom2###DefaultConfig
- AgentFrontend::TicketList::Custom3###DefaultConfig
- AgentFrontend::TicketList::Custom4###DefaultConfig
- AgentFrontend::TicketList::Custom5###DefaultConfig

`BusinessObjectListType` Type of the business object list to use.

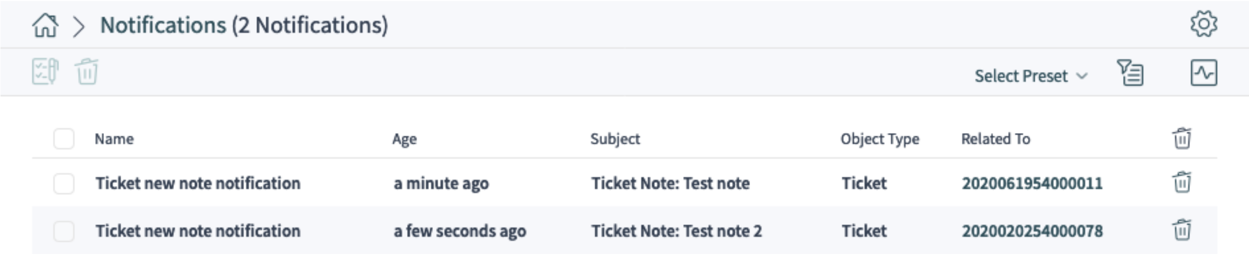
Possible values for the `BusinessObjectListType` key:

```
TicketList
KnowledgeBaseArticleList
```

SlugName Determines the search engine-friendly URL part via which the screen is available.

> **Note:** Currently only single word search engine-friendly URL parts are supported. Also note that the first character of the search engine-friendly URL part will always be converted to lower case variant, for example:
>
> ```
> AgentFrontend::TicketList::Static => /agent/tickets/static
> AgentFrontend::TicketList::Unresolved => agent/tickets/unresolved
> AgentFrontend::TicketList::Reminders => agent/tickets/reminders
> AgentFrontend::KnowledgeBaseArticleList::Added => agent/knowledge-
>  ↪base-articles/added
> ```

AgentFrontend::WebNotificationList###DefaultConfig The default configuration for the *Web Notifications* list.



	Name	Age	Subject	Object Type	Related To
□	Ticket new note notification	a minute ago	Ticket Note: Test note	Ticket	2020061954000011
□	Ticket new note notification	a few seconds ago	Ticket Note: Test note 2	Ticket	2020020254000078

Fig. 43: An Example of the *Web Notifications* List Screen

Link to the reference of the system configuration:

- AgentFrontend::WebNotificationList###DefaultConfig

AgentFrontend::Search###DefaultConfig The default configuration for the *Search Results* list.

Link to the reference of the system configuration:

- AgentFrontend::Search###DefaultConfig

AgentFrontend::TicketList::ArticlePreview###DefaultConfig The default configuration for the *Article Preview* in the *Ticket* list.

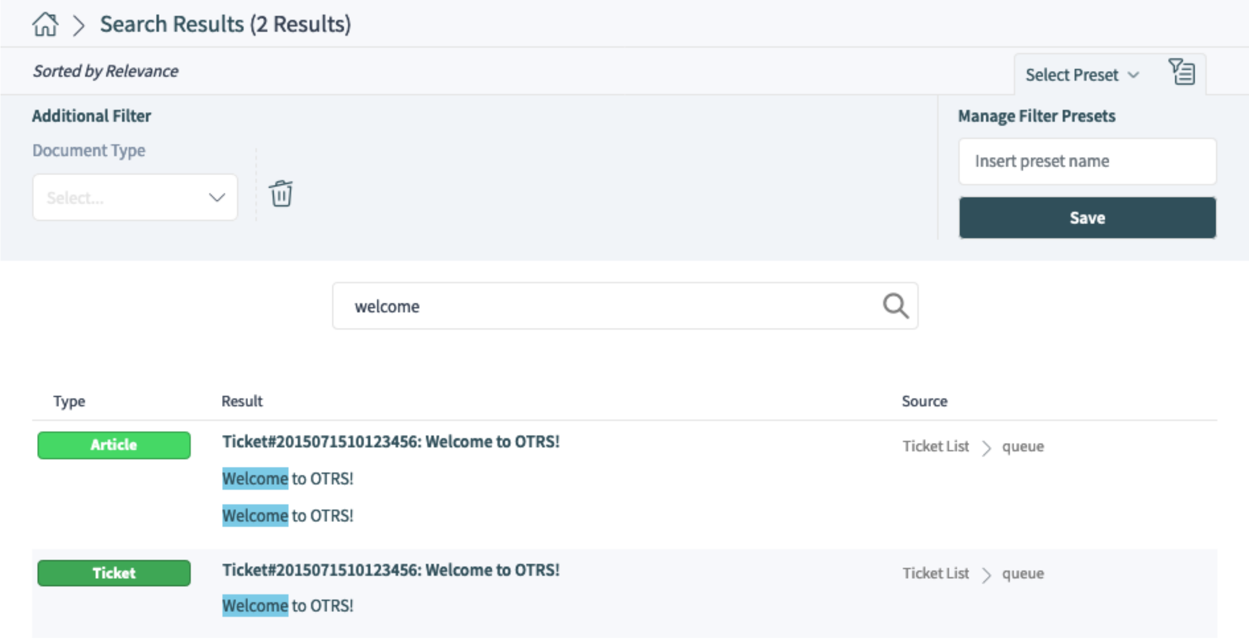
Link to the reference of the system configuration:

- AgentFrontend::TicketList::ArticlePreview###DefaultConfig

AgentFrontend::*###DefaultConfig The default configuration for the inline business object lists. Inline list tables are shown as part of several actions like *Merge Ticket*, *Link Objects* or *Append to Ticket*.

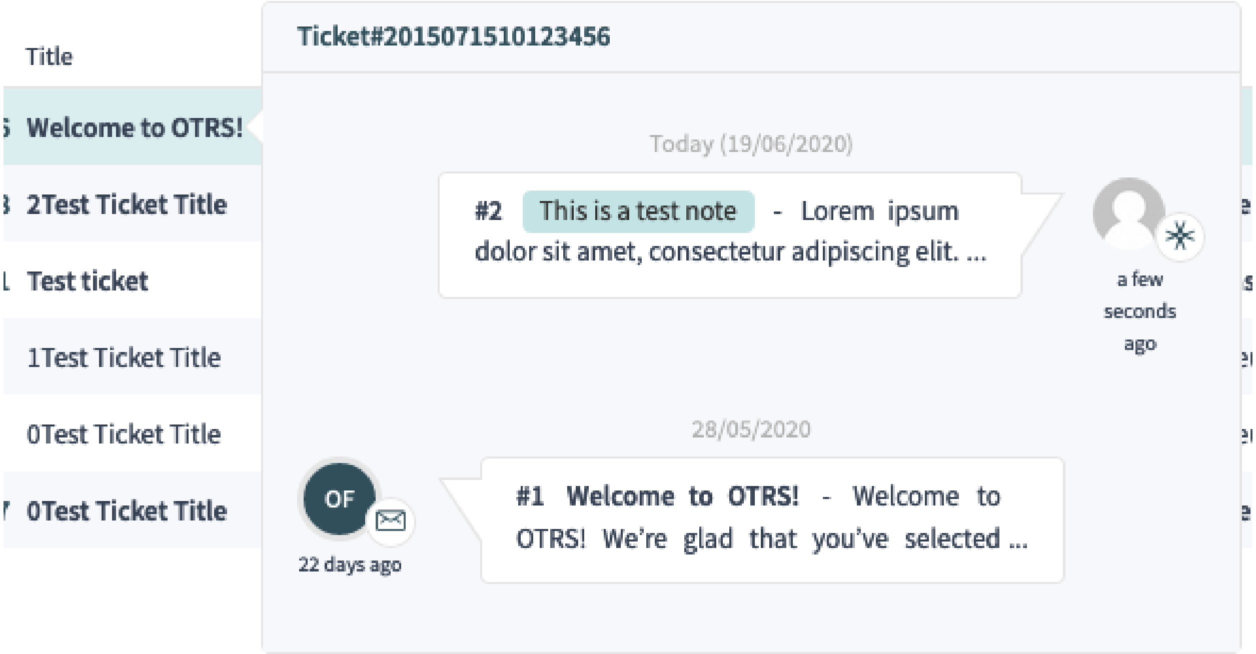
The following system configuration settings are relevant:

- AgentFrontend::Merge::Ticket###DefaultConfig
- AgentFrontend::LinkObject::Ticket###DefaultConfig
- AgentFrontend::LinkObject::KnowledgeBaseArticle###DefaultConfig
- AgentFrontend::LinkObject::CalendarAppointment###DefaultConfig
- AgentFrontend::Chat::AppendToTicket###DefaultConfig



Type	Result	Source
Article	Ticket#2015071510123456: Welcome to OTRS! Welcome to OTRS! Welcome to OTRS!	Ticket List > queue
Ticket	Ticket#2015071510123456: Welcome to OTRS! Welcome to OTRS!	Ticket List > queue

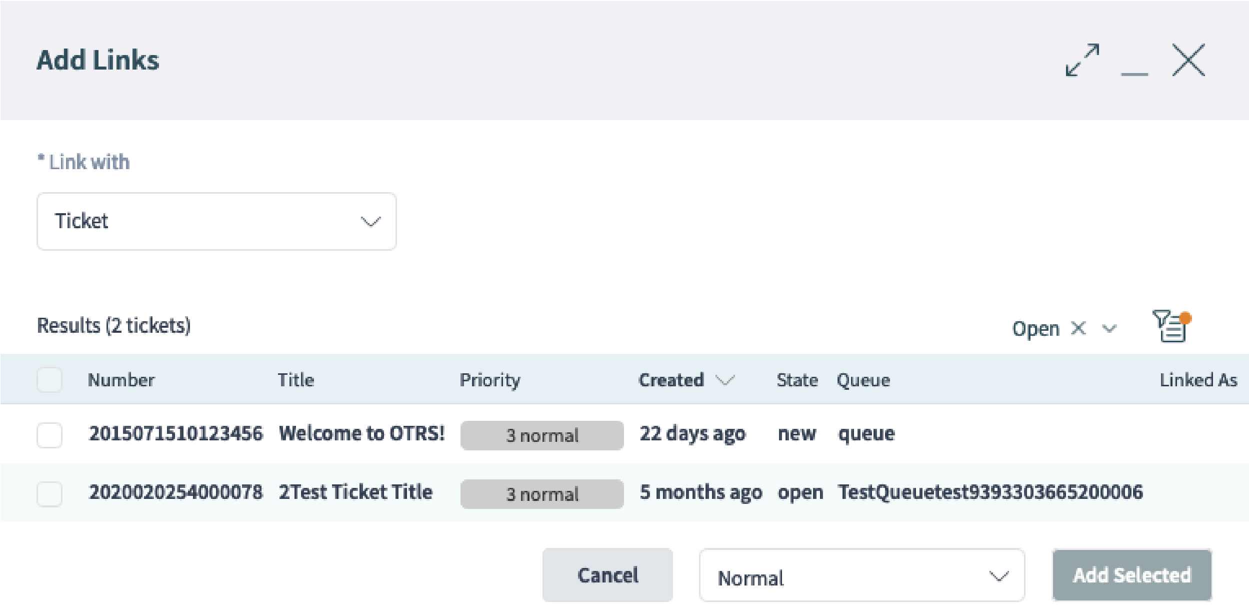
Fig. 44: An Example of the *Search Results* List Screen



#2 This is a test note - Lorem ipsum dolor sit amet, consectetur adipiscing elit. ...

#1 Welcome to OTRS! - Welcome to OTRS! We're glad that you've selected ...

Fig. 45: An Example of the *Article Preview* in the Ticket List Screen



Add Links

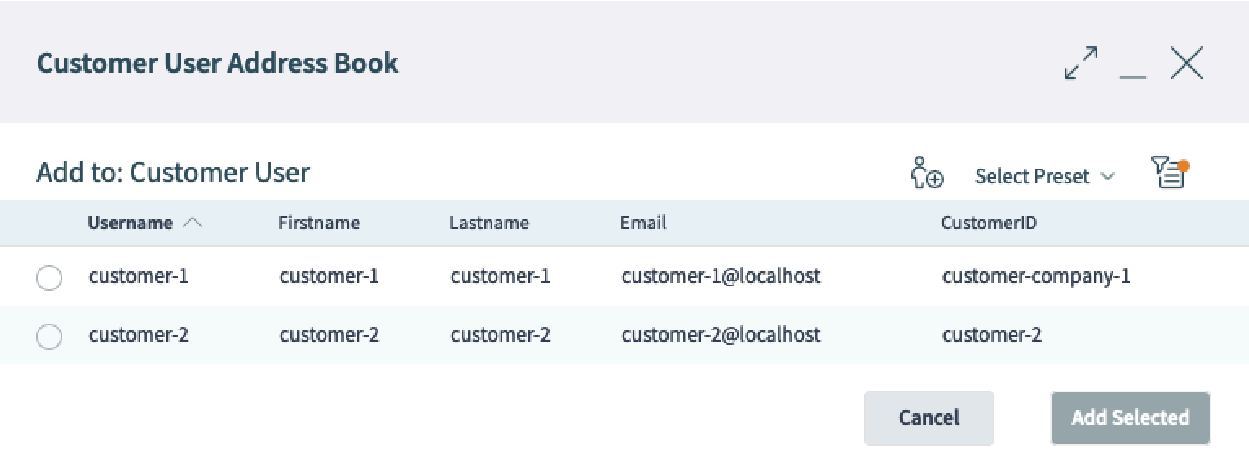
* Link with: Ticket

Results (2 tickets)

□	Number	Title	Priority	Created	State	Queue	Linked As
□	2015071510123456	Welcome to OTRS!	3 normal	22 days ago	new	queue	
□	2020020254000078	2Test Ticket Title	3 normal	5 months ago	open	TestQueuetest9393303665200006	

Normal

Fig. 46: An Example of the *Link Objects* Action in the Ticket Detail View



Customer User Address Book

Add to: Customer User

	Username	Firstname	Lastname	Email	CustomerID
○	customer-1	customer-1	customer-1	customer-1@localhost	customer-company-1
○	customer-2	customer-2	customer-2	customer-2@localhost	customer-2

Fig. 47: An Example of the *Customer User Address Book* List Screen

`AgentFrontend::*AddressBookList*###DefaultConfig` The default configuration for the *Customer User* and *Customer* address book screens. The address books allow contextual adding of customer and customer user records to target form fields.

The following system configuration settings are relevant:

- AgentFrontend::CustomerUserAddressBookList::Email###DefaultConfig
- AgentFrontend::CustomerUserAddressBookList::SMS###DefaultConfig
- AgentFrontend::CustomerCompanyAddressBookList###DefaultConfig

Each business object list setting has a `Config` key which contains its configuration.

See also:

The detailed explanation of the possible `Config` key values can be found in the *Business Object List YAML Configuration* chapter.

YAML Configuration Basics

Before we delve too deeply into the business object configuration, it is important to explain the YAML syntax which is heavily used for some of the setting values.

YAML (a recursive acronym for *YAML Ain’ t Markup Language*) is a **human-readable** data-serialization language. It is commonly used for configuration files, but it can describe data structures of arbitrary complexity.

Although YAML supports many of them, only a few data types are used in the context of OTRS configuration:

Scalar This is the most basic data type; it could be a number, a string, or a boolean expression. Sometimes it is also referred to as a **value**.

```
Number: 42
String1: Foo
String2: 'Foo bar'
String3: "Foo bar Xyzzy"
Boolean1: 1
Boolean2: 0
```

Note: Scalar data type is never used on its own, but it is always part of another higher-level structure.

List List contains an arbitrary number of items which can be of any data type. Sometimes it is also referred to as an **array**.

Each item in a list is divided from others by a separator in the form of a minus sign followed by a single space.

```
List1:
- Item1
- Item2
- Item3
```

It is recommended that list items are indented via a minus sign followed by a single space. This style is used throughout this guide and default configuration.

Warning: In YAML, the number of spaces is important. The number of spaces used for indentation must always be consistent, otherwise unexpected structures or syntax errors can occur.

Note: To define an empty list, you can use the square bracket syntax:

```
EmptyList: []
```

Object Object is a data type consisting of key/value pairs, separated by the colon punctuation mark. Sometimes it is also referred to as a **hash**.

Object **key** is always on the left side of the colon punctuation mark, while the **value** is always on its right side (or indented below):

```
Object1:
  Key1: Value1
  Key2: Value2
  Key3: Value3
```

It is recommended that object key/value pairs are indented with two spaces. This style is used throughout this guide and default configuration.

Warning: In YAML, the number of spaces is important. The number of spaces used for indentation must always be consistent, otherwise unexpected structures or syntax errors can occur.

Note: To define an empty object, you can use the curly brackets syntax:

```
EmptyObject: {}
```

To signal that a text uses YAML syntax, it is recommended to add a **document header** to its first line that consists of three consecutive minus signs, followed by the line break. By doing this consistently, it becomes very easy to identify YAML structures when mixed with other configurations.

With the three basic data types explained above, it is possible to create more complex structures. For example, an **array of hashes**:

```
---
Array:
- Key1: A
  Key2: B
  Key3: C
- Key1: 1
  Key2: 2
  Key3: 3
- Key1: I
  Key2: II
  Key3: III
```

Note: In a list that contains multiple object keys, only the first key must be prefixed with the minus sign followed by a space. Each subsequent key must be prefixed with two spaces.

The following example illustrates the usage of a **hash of arrays**:

```
---
Hash:
  Key:
    Subkey1:
    - Item1
    - Item2
    - Item3
    Subkey2:
    - Item4
    - Item5
    - Item6
```

Note: If the object keys are separated by other structures, it is important to keep them on the same indentation level. Since they are all siblings, they must be prefixed by the same number of spaces.

It is also possible to mix the data types that result in more complex data structures:

```
---
Key1:
- Subkey1: Value1
- Subkey2: Value2
Key2: Value3
Key3:
  Subkey3:
  - Value4
  - Value5
```

YAML supports comments in its code, so it is possible to include additional text that will be ignored during parsing. This is useful in case users need to describe certain parts of the structure, perhaps for future reference, or to quickly ignore part of the code without actually removing it.

Comments can be marked with `#`. Any following text will be ignored.

```
---
# This is a comment
Key: Value # This is another comment
# Key2: Value 2
```

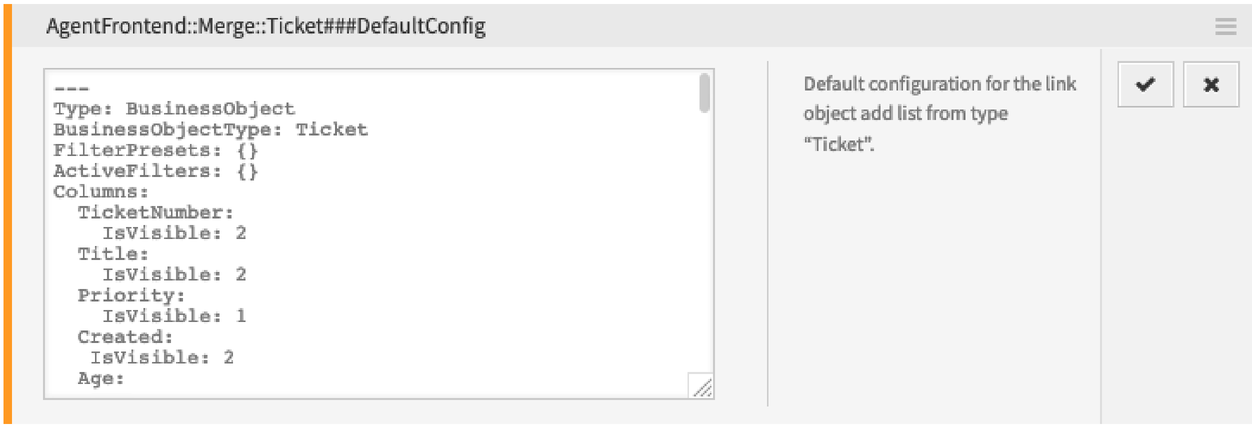
In the example above, the last line will be ignored too since it starts with `#`.

If you actually want to use the pound sign as part of your string of values, simply enclose the complete string in quotes.

```
---
String: 'Comments start with # character'
```

When you edit an OTRS system setting with the YAML value type, a suitable text editor will be displayed.

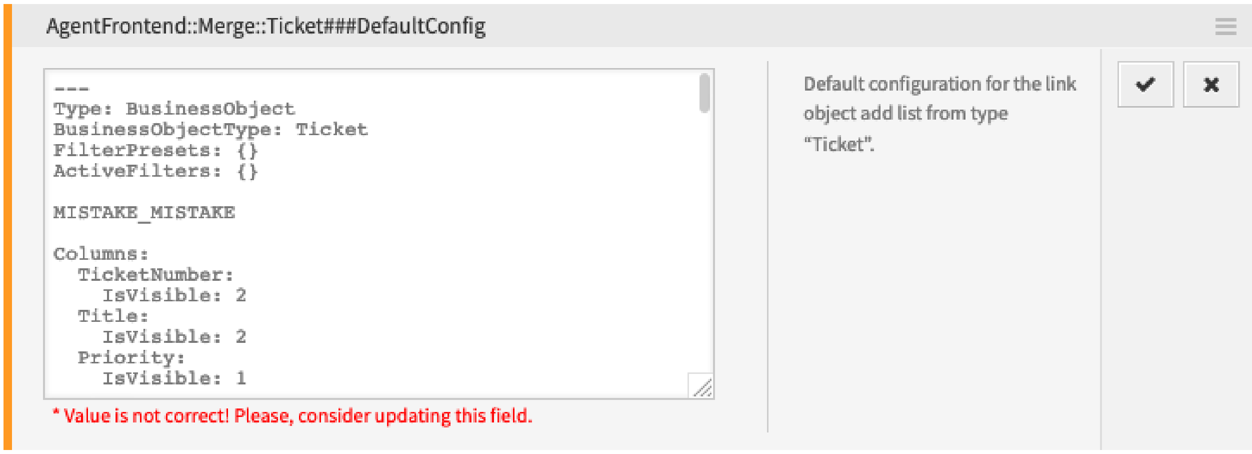
When saving the setting, the YAML structure will be checked for validity. In case of an error, it is possible to change and correct the structure.


AgentFrontend::Merge::Ticket###DefaultConfig

```
---
Type: BusinessObject
BusinessObjectType: Ticket
FilterPresets: {}
ActiveFilters: {}
Columns:
  TicketNumber:
    IsVisible: 2
  Title:
    IsVisible: 2
  Priority:
    IsVisible: 1
  Created:
   IsVisible: 2
  Age:
```

Default configuration for the link object add list from type "Ticket".

Fig. 48: System Setting with a YAML Value


AgentFrontend::Merge::Ticket###DefaultConfig

```
---
Type: BusinessObject
BusinessObjectType: Ticket
FilterPresets: {}
ActiveFilters: {}

MISTAKE_MISTAKE

Columns:
  TicketNumber:
    IsVisible: 2
  Title:
    IsVisible: 2
  Priority:
    IsVisible: 1
```

* Value is not correct! Please, consider updating this field.

Default configuration for the link object add list from type "Ticket".

Fig. 49: Error in a YAML Structure

It is important to understand that only the syntax of the structure is checked during validation. A deep validation will not occur. It is your responsibility as an administrator to ensure that the configuration you specify is correct.

> **Warning:** The most common errors that occur are related to indentation. You have to ensure that your indentations are consistent, especially the indentations of the different data types. For example, the following two structures are considered different.
>
> ```
> ---
> Object:
>    Key:
>       Subkey: Value
> ```
>
> ```
> ---
> Object:
>    Key:
>    Subkey: Value
> ```
>
> Since the `Subkey` in the second example is not aligned properly within its parent `Key`, it is considered to be its sibling part and not a child. In this case the actual value of the `Key` will be empty.

Business Object List YAML Configuration

A business object list YAML configuration can have many parameters. The following example shows how to construct a business object list in general.

```
Type: BusinessObject
BusinessObjectType: Ticket
ScreenTitle: <Some Screen Title>
Changeable: 1
FilterPresets:
  "<Displayed Filter Preset Name>":
    <FilterName1>:
      Value: <Value>
    <FilterName2>:
      Value:
      - <Value1>
      - <Value2>
    <FilterName3>:
      Value:
        <Parameter1>: <Value1>
        <Parameter2>: <Value2>
  ...
ActiveFilters:
  <FilterName1>:
    Value: <Value>
  <FilterName2>:
    Value:
    - <Value1>
    - <Value2>
  <FilterName3>:
    Value:
```

(continues on next page)

(continued from previous page)

```
      <Parameter1>: <Value1>
      <Parameter2>: <Value2>
      ...
  ...
AllowGETConfig:
- <ConfigurationName1>
- <ConfigurationName2>
- ...
Columns:
  <ColumnName1>:
    IsVisible: 0|1|2
    IsInlineEditable: 0|1
  ...
DefaultColumnOrder:
- <ColumnName1>
- <ColumnName2>
- ...
HideAvailableFilters:
- <FilterName1>
- <FilterName2>
- ...
ItemsPerPage: 10
Limit: 1000
SortBy:
- Column: <ColumnName>
  Direction: Down|Up
- ...
AvailableDynamicFieldFilters:
- <FilterName1>
- <FilterName2>
- ...
```

The following YAML keys and values can be used for business object lists.

`Type` Defines the type of the setting. This is an internal value which is always set to `BusinessObject` and must not be changed.

> **Warning:** Changing this value might result in a broken system configuration.

`BusinessObjectType` Defines the business object type of the business object list. This is an internal value which is specific for the current list type and must not be changed.

> **Warning:** Changing this value might result in a broken system configuration.

`ScreenTitle` Defines the title of the business object list.

> **Note:** In some places, you may notice `# translatable` comments that are displayed next to the text values. This marker is only used for the internal collection of translatable strings and has no effect in the changed settings. Adding this comment to modified values will not result in the actual text translation.

The visible text in the configuration can be translated into other languages with custom translation files. For more information, see the *Custom Translation File* chapter in the developer manual.

`Changeable` Defines whether the business object list can be personalized by the user in the front end. If turned off, the view configuration will not be available and also filter presets for the view cannot be saved. The value can be `1` (on) or `0` (off). The assumed default is off when the key is not used in the definition.

`FilterPresets` Defines the pre-defined filters used for the business object list. Multiple filters can be defined here. The filter value can be a string, array or a hash. The following YAML snippet shows examples for all types.

```
FilterPresets:
  Closed:
    StateType:
      Value: Closed
  Locked:
    LockIDs:
      Value:
      - 2
  "Reminder Reached":
    TicketPending_DateTimeRelative:
      Value:
        Format: minute
        Point: 1
        Start: Before
```

See also:

The list of possible filter names can be found in the *Filter Names* chapter.

`ActiveFilters` Defines the filters that are active when displaying the business object list. Multiple filters can be added here; each is defined by the filter name. The filter value can be a string, array or a hash. The following YAML snippet shows examples for both.

```
ActiveFilters:
  StateType:
    Value: Closed
  TicketClose_DateTimeRelative:
    Value:
      Start: Before
      Point: 1
      Format: minute
```

See also:

The list of possible filter names can be found in the *Filter Names* chapter.

`AllowGETConfig` Defines what parameters can be changed via the URL with the `Config` URL query parameter:

```
AllowGETConfig:
- VisibleColumns
- SortBy
- ActiveFilters
- FilterPresets
```

(continues on next page)

(continued from previous page)

```
- ItemsPerPage
- FilterPresetSelected
```

The list should contain names of other configuration keys for which definition will be allowed when passed via the correct URL query parameter to the view.

See also:

For an example and more information, please see the section about *Custom URL Support*.

`Columns` Defines the columns that are available for the business object list in the front end.

> **`IsVisible`** Defines whether the column is not visible (`0`), not visible by default but the agent can make it visible (`1`) or visible by default (`2`).
>
> **`IsInlineEditable`** Defines whether the value in the column is inline editable (`1`) or not (`0`).
>
> **See also:**
>
> The list of possible column names can be found in the *Column Names* chapter.

`DefaultColumnOrder` Defines the default column order in the business object list.

`HideAvailableFilters` Defines the filters which are not available for the agents in the front end, either in the view configuration or in the filter presets.

`ItemsPerPage` Defines the number of objects per page that are displayed by default and the number of new objects that are loaded when the agent scrolls down the list. If omitted, it takes the default value of `10`. Although there is no upper limit, it is not advisable to set the value greater than `100` for performance reasons.

`Limit` Defines the maximum amount of displayed objects in a business object list. Most of the list types in the system are limited to an upper boundary of 10,000 objects for performance reasons.

`SortBy` Defines the sorting criteria and the sorting order of the objects in the business object list.

`AvailableDynamicFieldFilters` Defines the list of dynamic fields by name which are available as filters.

Business Object Detail Views

The business object detail views can be organized into several families.

The following settings define the default column layout configuration for the detail view screens.

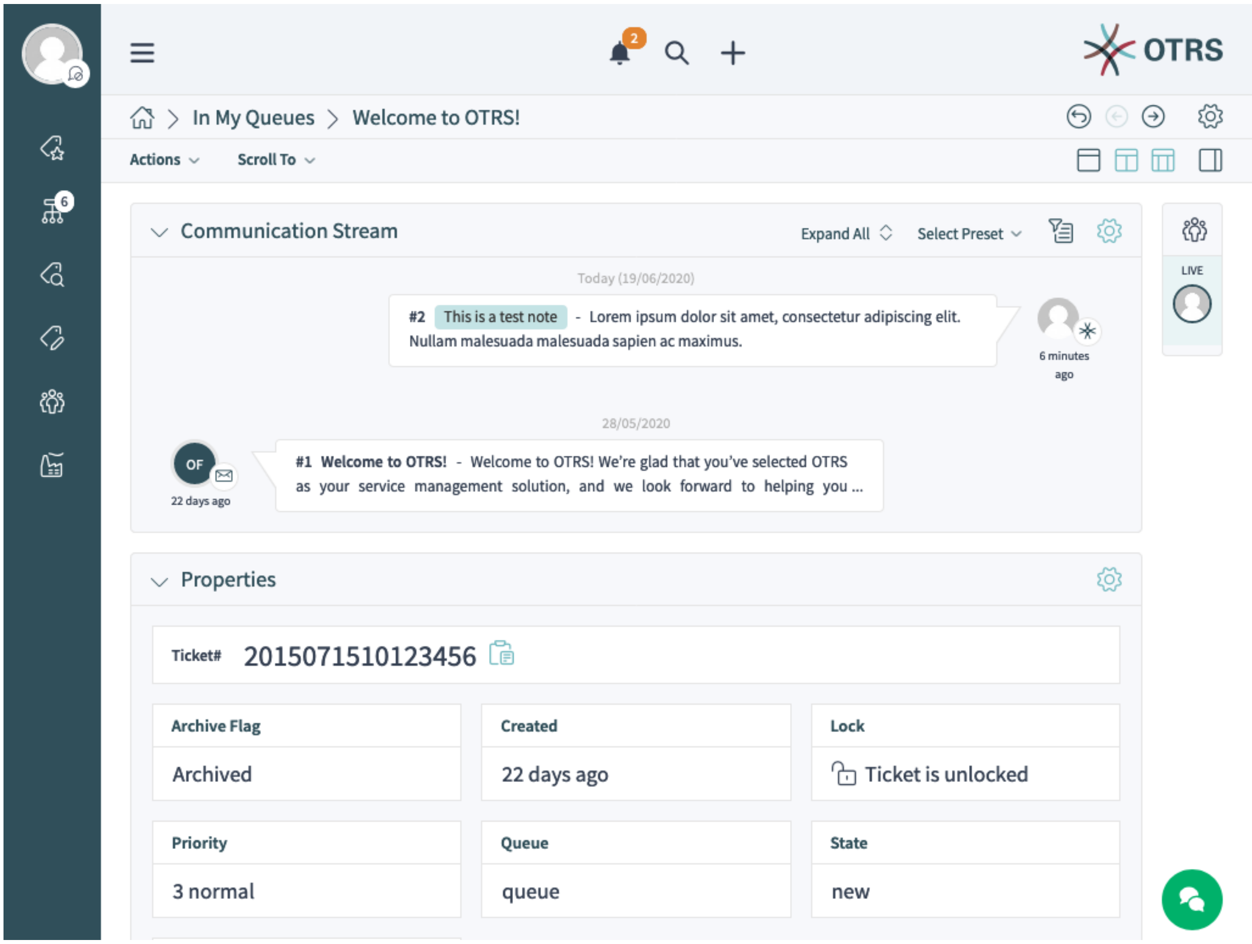
The following system configuration settings are relevant:

- AgentFrontend::CustomerCompanyDetailView###001-Framework
- AgentFrontend::CustomerUserDetailView###001-Framework
- AgentFrontend::KnowledgeBaseArticleDetailView###001-Framework
- AgentFrontend::TicketDetailView###001-Framework

The following settings define the default column layout configuration for the business object overview screens.

The following system configuration settings are relevant:

- AgentFrontend::CalendarOverview###001-Framework
- AgentFrontend::Dashboard###001-Framework



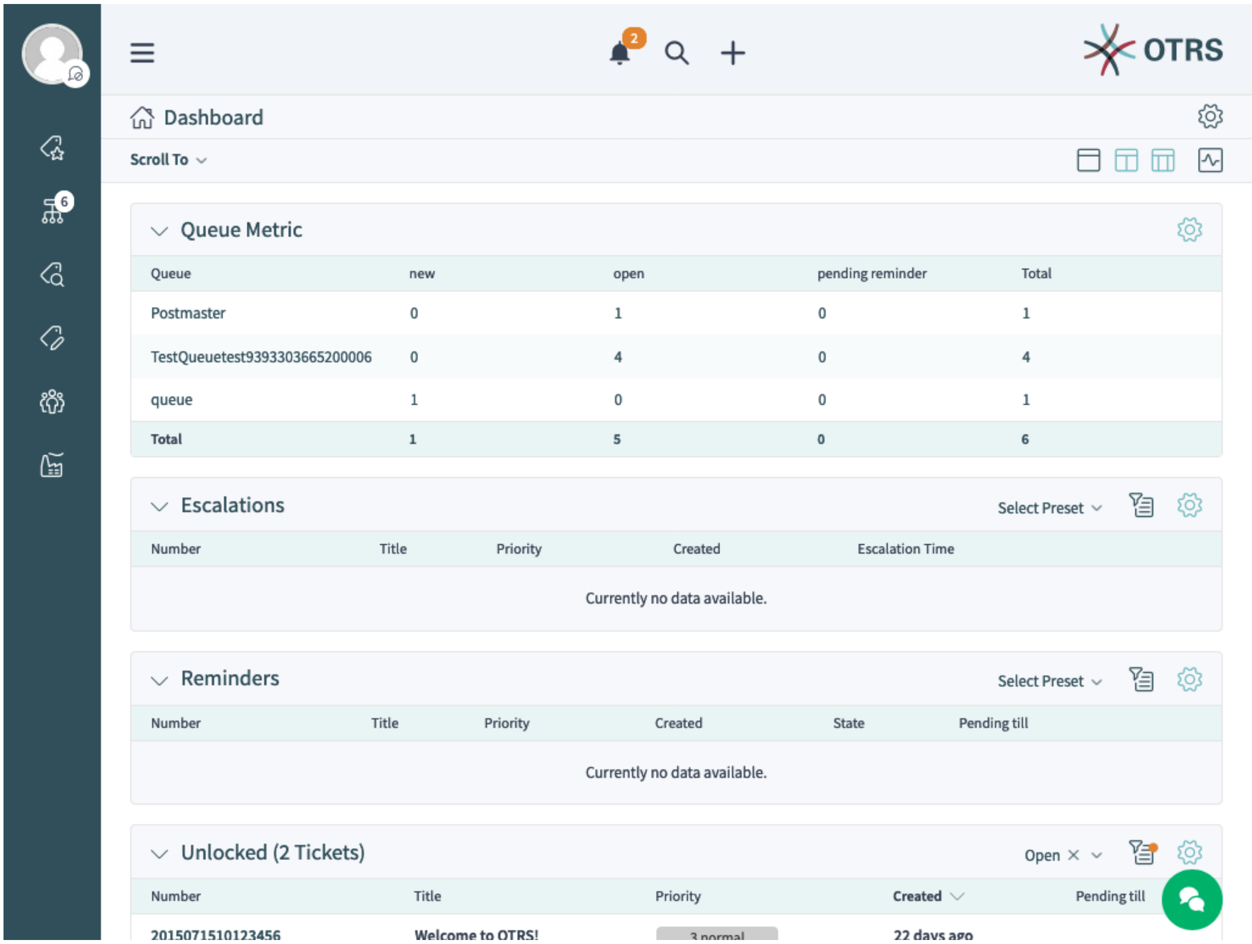
#2 This is a test note - Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam malesuada malesuada sapien ac maximus.

#1 Welcome to OTRS! - Welcome to OTRS! We're glad that you've selected OTRS as your service management solution, and we look forward to helping you ...

Ticket# 2015071510123456

Archive Flag	Created	Lock
Archived	22 days ago	Ticket is unlocked
Priority	**Queue**	**State**
3 normal	queue	new

Fig. 50: An Example of the *Ticket* Detail View



Queue Metric

Queue	new	open	pending reminder	Total
Postmaster	0	1	0	1
TestQueuetest9393303665200006	0	4	0	4
queue	1	0	0	1
Total	**1**	**5**	**0**	**6**

Escalations

Number	Title	Priority	Created	Escalation Time

Currently no data available.

Reminders

Number	Title	Priority	Created	State	Pending till

Currently no data available.

Unlocked (2 Tickets)

Number	Title	Priority	Created	Pending till
2015071510123456	Welcome to OTRS!	3 normal	22 days ago	

Fig. 51: An Example of the *Dashboard* Screen

- AgentFrontend::StatisticReportOverview###001-Framework

The following settings define the default column layout configuration for the business object create screens.

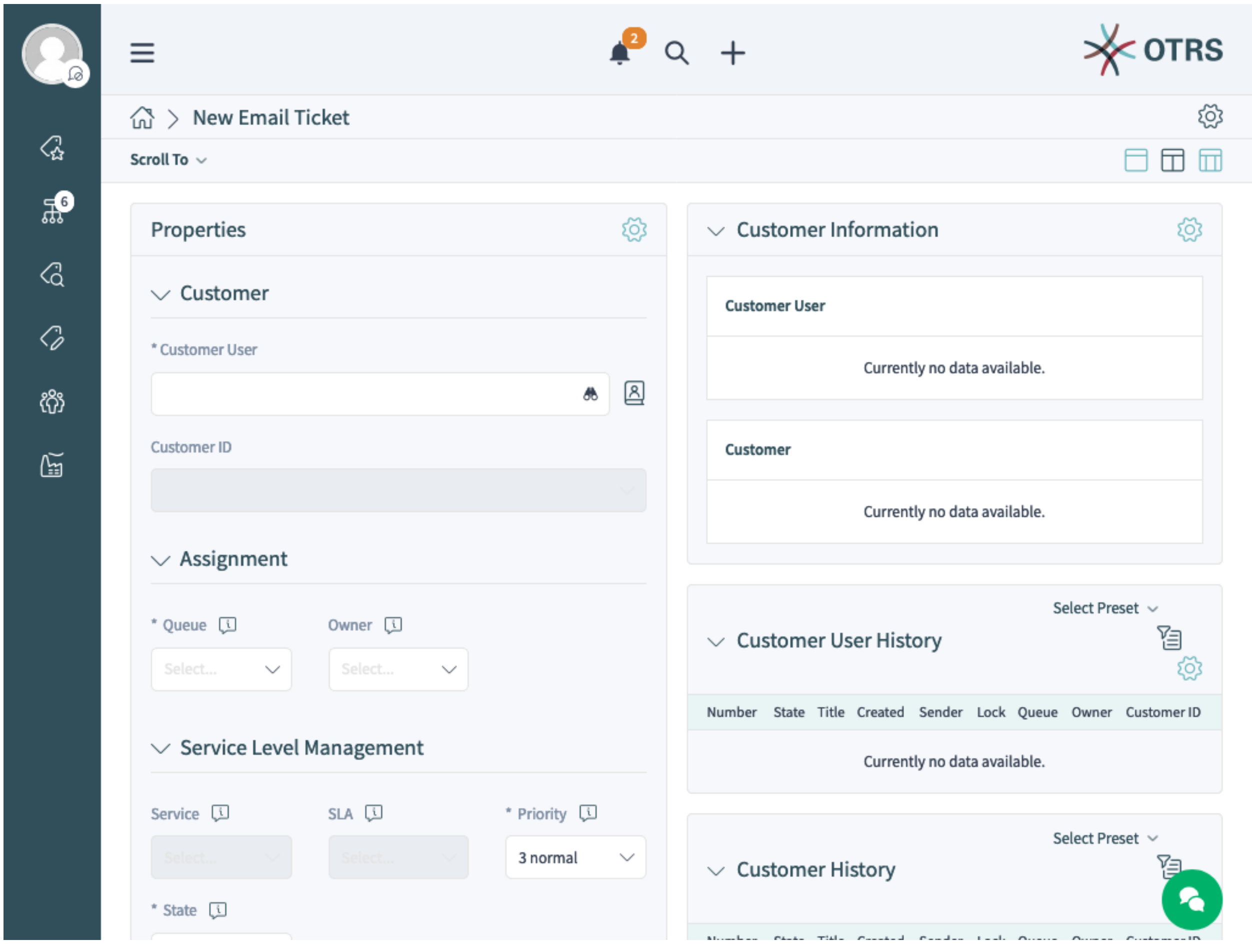


Fig. 52: An Example of the *Create Email Ticket* Screen

The following system configuration settings are relevant:

- AgentFrontend::CustomerCompanyCreate###001-Framework
- AgentFrontend::CustomerUserCreate###001-Framework
- AgentFrontend::KnowledgeBaseArticleCreate###001-Framework
- AgentFrontend::StatisticCreateUpdateView###001-Framework
- AgentFrontend::StatisticReportCreateUpdateView###001-Framework
- AgentFrontend::TicketCreate::Email###001-Framework
- AgentFrontend::TicketCreate::Phone###001-Framework
- AgentFrontend::TicketCreate::Process###001-Framework
- AgentFrontend::TicketCreate::SMS###001-Framework

The following settings define the custom column layout configurations for adding additional widgets to a specific view. By default, these are empty settings. If additional widgets are added to them, they will be merged with the remaining settings for the corresponding view.

The following system configuration settings are relevant:

- AgentFrontend::CalendarOverview###100-Custom
- AgentFrontend::CustomerCompanyCreate###100-Custom
- AgentFrontend::CustomerCompanyDetailView###100-Custom
- AgentFrontend::CustomerUserCreate###100-Custom
- AgentFrontend::CustomerUserDetailView###100-Custom
- AgentFrontend::Dashboard###100-Custom
- AgentFrontend::KnowledgeBaseArticleCreate###100-Custom
- AgentFrontend::KnowledgeBaseArticleDetailView###100-Custom
- AgentFrontend::StatisticCreateUpdateView###100-Custom
- AgentFrontend::StatisticReportCreateUpdateView###100-Custom
- AgentFrontend::StatisticReportOverview###100-Custom
- AgentFrontend::TicketCreate::Email###100-Custom
- AgentFrontend::TicketCreate::Phone###100-Custom
- AgentFrontend::TicketCreate::Process###100-Custom
- AgentFrontend::TicketCreate::SMS###100-Custom
- AgentFrontend::TicketDetailView###100-Custom

Each of the above settings has the same key structure:

`Type` Type of the business object screen.

Possible values for the `Type` key:

```
BusinessObjectCreate
BusinessObjectDetailView
BusinessObjectOverview
```

`BusinessObjectType` Business object type behind the screen.

Possible values for the `BusinessObjectType` key:

```
CustomerCompany
CustomerUser
Dashboard
KnowledgeBaseArticle
Statistic
StatisticReport
Ticket
```

`ColumnLayout` The YAML configuration for different column views.

See also:

The detailed explanation of the `ColumnLayout` key can be found in the *Business Object Detail View YAML Configuration* chapter.

Business Object Detail View YAML Configuration

The YAML structure defines the content for the one, two or three column views. The definition contains the information about which widgets are displayed per default in each view.

```
 OneColumn:
   1:
   - Name: <WidgetName1>
   - Name: <WidgetName2>
   ...
 TwoColumns:
   1:
   - Name: <WidgetName1>
   - Name: <WidgetName2>
   ...
   2:
   - Name: <WidgetName3>
   - Name: <WidgetName4>
   ...
 ThreeColumns:
   1:
   - Name: <WidgetName1>
   - Name: <WidgetName2>
   ...
   2:
   - Name: <WidgetName3>
   - Name: <WidgetName4>
   ...
   3:
   - Name: <WidgetName5>
   - Name: <WidgetName6>
   ...
StripeSidebar:
- Name: StripePeople
...
```

Note: The sidebar is only available for business objects of type `Ticket`.

Available widget names depend on the specific business object screen.

`CalendarOverview` Possible widget names for the *Calendar Overview* screen.

```
AppointmentsToday
AppointmentsThisWeek
AppointmentsThisMonth
```

`CustomerCompanyCreate` Possible widget names for the *Add Customer* screen.

```
CreateProperties
```

`CustomerCompanyDetailView` Possible widget names for the *Customer* detail view screen.

```
CustomerInformation
CustomerUserList
EscalatedTickets
OpenTickets
ReminderTickets
TicketList
```

`CustomerUserCreate` Possible widget names for the *Add Customer User* screen.

```
CreateProperties
```

`CustomerUserDetailView` Possible widget names for the *Customer User* detail view screen.

```
CustomerInformation
EscalatedTickets
OpenTickets
ReminderTickets
TicketList
```

`Dashboard` Possible widget names for the *Dashboard* screen.

```
CalendarView
CustomerList
CustomerUserList
DashboardIframe
DashboardImage
DashboardPeople
EscalatedTickets
KnowledgeBaseArticleList
News
OpenTickets
QueueOverview
RecentlyUpdatedKnowledgeBaseArticles
ReminderTickets
TicketList
UnlockedTickets
```

`KnowledgeBaseArticleCreate` Possible widget names for the *Add Knowledge Base Article* screen.

```
CreateProperties
```

`KnowledgeBaseArticleDetailView` Possible widget names for the *Knowledge Base Article* detail view screen.

```
KBAAttachments
KBAItemField1
KBAItemField2
KBAItemField3
KBAItemField4
KBAItemField5
KBAItemField6
KBALinkedObjects::CalendarAppointment
KBALinkedObjects::KnowledgeBaseArticle
KBALinkedObjects::Ticket
```

(continues on next page)

(continued from previous page)

```
KBAProperties
KBARating
People
```

`StatisticCreateUpdateView` Possible widget names for the *Create Statistic* and *Edit Statistic* screens.

```
CreateUpdateProperties
```

`StatisticReportCreateUpdateView` Possible widget names for the *Create Report* and *Edit Report* screens.

```
CreateUpdateProperties
```

`StatisticReportOverview` Possible widget names for the *Statistics and Reports* overview screen.

```
StatisticLists
StatisticMetrics
StatisticReportList
StatisticStatic
```

`TicketCreate::Email` Possible widget names for the *Create Email Ticket* screen.

```
CreateProperties
CustomerHistory
CustomerInformation
CustomerUserHistory
```

`TicketCreate::Phone` Possible widget names for the *Create Phone Ticket* screen.

```
CreateProperties
CustomerHistory
CustomerInformation
CustomerUserHistory
```

`TicketCreate::SMS` Possible widget names for the *Create SMS Ticket* screen.

```
CreateProperties
CustomerHistory
CustomerInformation
CustomerUserHistory
```

`TicketCreate::Process` Possible widget names for the *Create Process Ticket* screen.

```
CreatePropertiesProcess
ProcessInformation
```

`TicketDetailView` Possible widget names for the ticket detail view screen.

```
Attachments
BusinessProcessInformation
CommunicationCompact
CommunicationStream
CustomerInformation
```

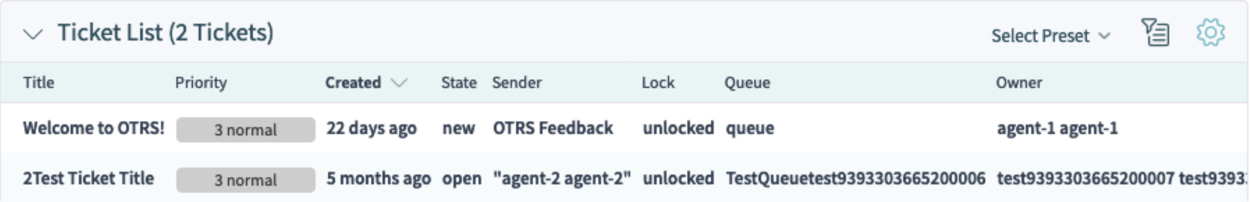
(continues on next page)

(continued from previous page)

```
FormDrafts
LinkedObjects::CalendarAppointment
LinkedObjects::KnowledgeBaseArticle
LinkedObjects::Ticket
People
Properties
```

Widget Type Definitions

Each widget type provides its own definition that will be inherited by any widget that uses it. For example, let’ s take a look how this definition looks for one of the dashboard widget types:



Title	Priority	Created	State	Sender	Lock	Queue	Owner
Welcome to OTRS!	3 normal	22 days ago	new	OTRS Feedback	unlocked	queue	agent-1 agent-1
2Test Ticket Title	3 normal	5 months ago	open	"agent-2 agent-2"	unlocked	TestQueuetest9393303665200006	test9393303665200007 test9393

Fig. 53: An Example of the *Ticket List* Widget Type in the *Dashboard* Screen

The configuration of this widget type is located in the AgentFrontend::Dashboard::WidgetType###TicketList system configuration setting in YAML format.

A widget type definition contains the following general YAML keys:

```
Type: BusinessObject
BusinessObjectType: Ticket
ActiveFilters: {}
FilterPresets: {}
Columns: {}
Collapsed: 0
Hidden: 0
```

`Type` Defines whether the related widget type handles a business object or other data. `BusinessObject` is currently the only supported type, but this might be extended by installed packages.

`BusinessObjectType` This key defines the type of object the configuration is valid for. Based on this type, the front end and the back end will handle this object differently. New business object types may also be added by installed packages.

`ActiveFilters` This key defines the default filters that are active when displaying the business object list. It contains a hash of filters. A single filter also has a hash structure that normally contains a filter value. The value can be a string, array or hash for special filters like dates.

```
ActiveFilters:
  <FilterName1>:
    Value:
      <Value>
  <FilterName2>:
      Value:
        <Value>
```

The following example illustrates two active filters which are showing only tickets that were closed recently:

```
ActiveFilters:
  StateType:
    Value: Closed
  TicketClose_DateTimeRelative:
    Value:
      Start: Before
      Point: 1
      Format: minute
```

See also:

The list of possible filter names can be found in the *Filter Names* chapter.

FilterPresets A hash of default filter preset names with a defined set of filters.

```
FilterPresets:
  "<Filter Preset Name 1>":
      <FilterName1>:
         Value: <Value>
  "<Filter Preset Name 2>":
      <FilterName2>:
         Value: <Value>
```

The following example illustrates three separate filter presets: locked, unlocked and unread tickets.

```
FilterPresets:
  Locked:
    LockIDs:
      Value:
      - 2
  Unlocked:
    LockIDs:
      Value:
      - 1
  Unread:
    AgentTicketFlagSeen:
      Value: Unread
```

See also:

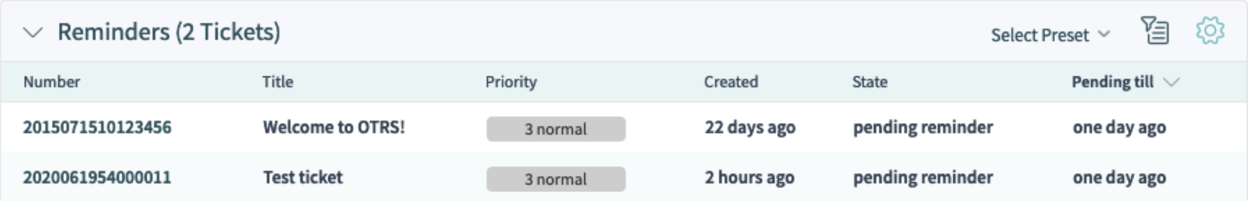
The list of possible filter names can be found in the *Filter Names* chapter.

Widget Definitions

Each widget provides its own definition that will be merged with the configuration of the inherited widget type. For example, let's take a look at what this definition looks like for one of the widgets in the customer business object detail view:

The configuration of this widget is located in the AgentFrontend::CustomerCompanyDetailView::Widget###ReminderTickets system configuration setting in YAML format.

A widget definition contains the following general YAML keys:



Reminders (2 Tickets)

Number	Title	Priority	Created	State	Pending till
2015071510123456	Welcome to OTRS!	3 normal	22 days ago	pending reminder	one day ago
2020061954000011	Test ticket	3 normal	2 hours ago	pending reminder	one day ago

Fig. 54: An Example of the *Reminder Tickets* Widget in the *Customer* Detail View

```
Title: Reminders # Translatable
Active: 1
IsVisible: 1
IsAlwaysPresent: 0
IsDuplicatable: 1
Config: {}
```

Active Defines whether the widget is active.

Collapsed Defines whether the form group is collapsed by default.

Config This key contains the complete default configuration of a widget. If left empty, the configuration of the widget type is used.

Hidden Defines whether the widget is hidden.

IsVisible Defines whether the widget is visible per default.

IsAlwaysPresent Defines whether the widget can be removed from a view by the user.

IsDuplicatable Defines whether the widget can be placed in a view multiple times.

Title Defines the displayed title of a widget in the header section.

> **Note:** In some places, you may notice `# translatable` comments that are displayed next to the text values. This marker is only used for the internal collection of translatable strings and has no effect in the changed settings. Adding this comment to modified values will not result in the actual text translation.
>
> The visible text in the configuration can be translated into other languages with custom translation files. For more information, see the *Custom Translation File* chapter in the developer manual.

If the widget contains a business object list:

Columns A hash with column names that are displayed in the business object list.

```
Columns:
  <ColumnName1>:
     IsVisible: <Value>
  <ColumnName2>:
     IsVisible: <Value>
```

Possible values for the `IsVisible` key:

- `0` = not available
- `1` = available but not visible
- `2` = available and visible

See also:

The list of possible column names can be found in the *Column Names* chapter.

ArticleDynamicFields Contains a list of dynamic fields to be displayed in the article preview. The prefix DynamicField_ must be added, for example DynamicField_MyFieldName.

```
ArticleDynamicFields:
  - <DynamicField_FieldName1>
  - <DynamicField_FieldName2>
  ...
```

ArticleViewType Defines whether the articles should be displayed collapsed or expanded by default.

```
ArticleViewType: collapsed
```

DefaultColumnOrder An array of column names that defines the default column order in the business object list.

```
DefaultColumnOrder:
- <ColumnName1>
- <ColumnName2>
...
```

DefaultFilterPresetFields Defines the default filter preset fields and their values.

```
DefaultFilterPresetFields:
  <FilterName1>:
    Value: '<Value>'
  <FilterName2>:
    Value: '<Value>'
```

CountPolling This determines whether the organizer item queries the current number of its objects in the background and displays them in a speech bubble next to the icon.

```
CountPolling: ShowNumberFoundItems
```

Possible values for the CountPolling key:

- 0 = do not show number
- ShowNumberFoundItems = show number of found tickets

Header

This key contains the definition of information fields to be displayed in the header section of the agent business cards.

```
Header:
  Properties:
  - Name: Avatar
    IsVisible: 1
```

HideAvailableFilters This key contains a list of filter names that are not present in the list of available filters in the widget configuration.

```
HideAvailableFilters:
- <FilterName1>
- <FilterName2>
- <FilterName3>
```

The following example hides filters for the ticket customer, text search and ticket type:

```
HideAvailableFilters:
- CustomerID
- Fulltext
- TypeIDs
```

`Identifier` Defines the property that will be used as a full-width card in the *Properties* widget. The property must be a unique identifier of the business object. If defined this property can be configured separately from other cards.

```
Identifier:
  Name: TicketNumber
  IsVisible: 0
```

`InitialLimit` Defines the limit of displayed agents in the stripe sidebar in the collapsed state.

`ItemsPerPage` A number that defines the number of items per page or load.

`Limit` A number that defines the maximum amount of displayed items.

`LastUsedFilterPreset` Defines the last used filter preset of a widget. This will be automatically applied when the widget is loaded.

```
LastUsedFilterPreset: "Reminder Reached"
```

`Properties` The properties are special containers (cards) within a widget that contain detailed information about a specific value of the associated business object. The cards have the option to allow editing of the associated value directly, such as setting a new queue or the owner of a ticket. The properties list in the configuration contains the name and default visibility state of every property.

```
Properties:
- Name: ArchiveFlag
  IsVisible: 1
- Name: Created
  IsVisible: 1
- Name: CustomerTickets
  IsVisible: 1
- Name: Lock
  IsVisible: 1
  IsInlineEditable: 0
- Name: Watch
  IsVisible: 1
  IsInlineEditable: 0
```

Possible values for the `IsVisible` key:

- `0` = not available
- `1` = available but not visible
- `2` = available and visible

Possible values for the `IsInlineEditable` key:

- `0` = not editable
- `1` = editable

`ShowPropertyOnEmpty` Defines whether the customer or customer user business card is displayed in the customer information widget when it does not contain a value.

```
ShowPropertyOnEmpty: 1
```

Note: This key is supported only by the customer and customer user business cards and cannot be used for regular property cards.

`SortBy` Defines the sort order of the business object list. To sort by multiple columns simultaneously, add each sort column as a separate element in the configuration.

```
SortBy:
- Column: Created
  Direction: Down
- Column: Priority
  Direction: Up
```

See also:

The list of sortable columns per business object type can be found in the *Column Names* chapter. Look for the (sortable) marker next to the column name.

Note: Only certain business object types support the multi-sorting feature, please see *Column Names* chapter for more information.

Forms

The configurable forms can be used in many screens, including ticket and article actions. The form configuration will define the fields and properties displayed for each action. Here is the form of *Add Note* action as an example.

The configuration of this form is located in the Forms###AgentFrontend::Ticket::Action::Note system configuration setting in YAML format.

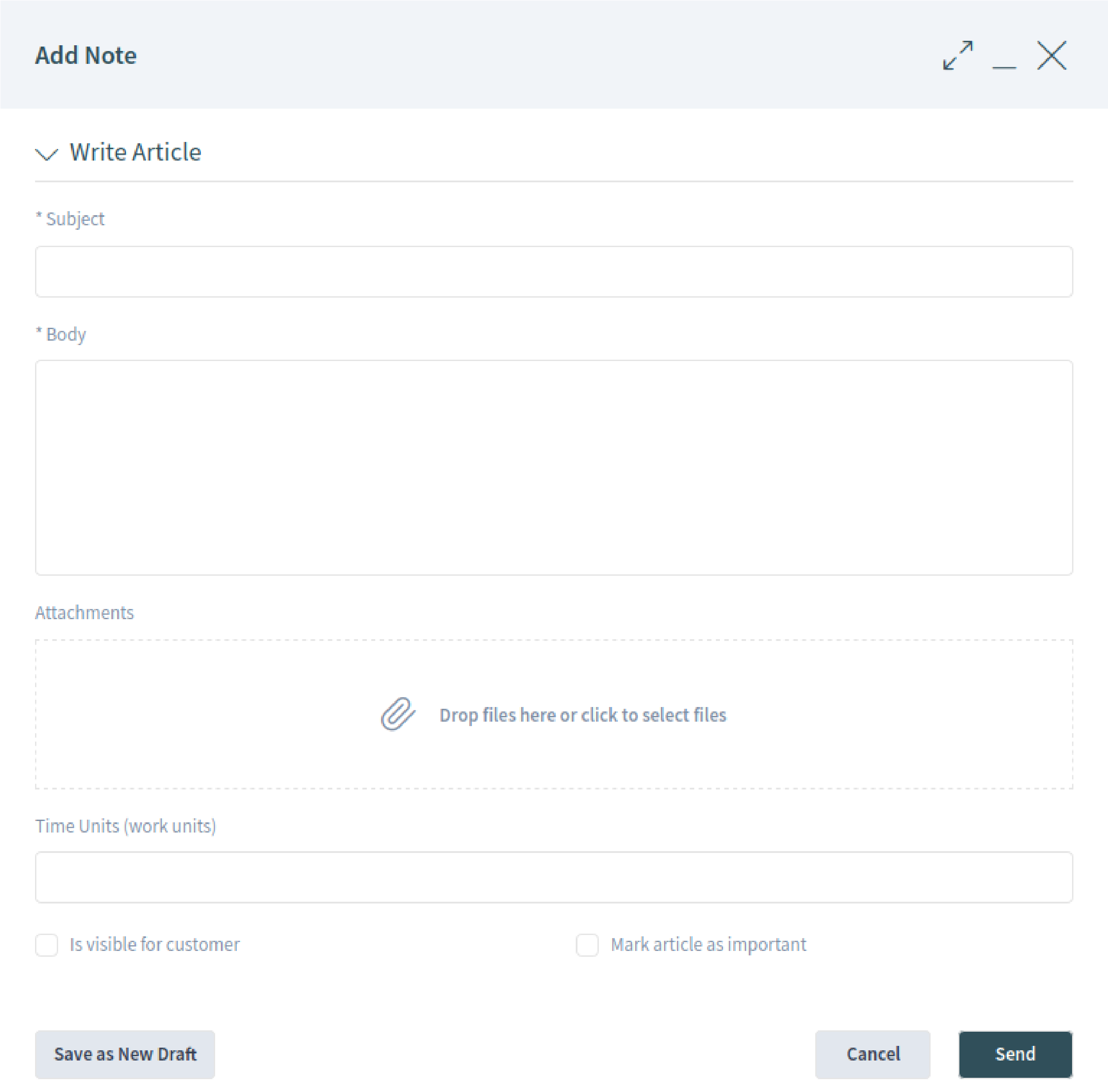
In general a form can contain a list of fields and form groups. Both are optional and can be added multiple times.

```
---
- <Field>
- ...
- <FormGroup>
- ...
```

The field is the basic element of a form. A field can have multiple properties.

```
- Name: <InternalName>
  Label: <Displayed Label>
```

(continues on next page)



Add Note

Write Article

* Subject

* Body

Attachments

Drop files here or click to select files

Time Units (work units)

□ Is visible for customer

□ Mark article as important

Fig. 55: Form of *Add Note* Action

(continued from previous page)

```
  Config:
    <Parameter>:
    - <value>
    - ...
  Default: <default value>
  Description: <some description>
  Disabled: 1
  Hidden: 1
  Hint: <some hint>
  Required: 1
```

The following keys and values can be used for the fields.

> **Warning:** The keys marked with the * (mandatory) symbol are mandatory. Skipping them might result in broken configuration and/or non-functional features.
>
> If a non-mandatory key is missing from the form field configuration, it will assume the built-in default behavior.

> **Warning:** Most forms and their fields are subject to existing *Access Control Lists (ACL)*. These rules have the final word on which values can be used in the fields and which will overwrite the form configuration.

Name * Defines the internal name of the field. This key has pre-defined values for each form.

See also:

The list of possible field names can be found in the *Form Fields* chapter.

The dynamic fields can also be added to the form by using the `DynamicField_` prefix and the name of the dynamic field. For example, if there is a dynamic field with the name `TestDropdown`, you need to use `DynamicField_TestDropdown` in the form configuration.

Label Defines the label of the field displayed above the input element. If omitted, the default label is displayed for the field specified with `Name`.

> **Note:** In some places, you may notice `# translatable` comments that are displayed next to the text values. This marker is only used for the internal collection of translatable strings and has no effect in the changed settings. Adding this comment to modified values will not result in the actual text translation.
>
> The visible text in the configuration can be translated into other languages with custom translation files. For more information, see the *Custom Translation File* chapter in the developer manual.

Config Defines the values that can be selected in a drop-down list. The values depend on the field specified with `Name`.

```
- Name: StateID
  Config:
    StateType:
    - open
    - pending auto
```

(continues on next page)

(continued from previous page)

```
      - pending reminder
      - closed
  Default: 4 # open
```

Note: The `Config` key is not applicable for all fields. It is only supported for specific fields. If the key is missing from the default configuration, it is most likely not supported by that field.

`Default` Defines the default value for the field. If the form field refers to an ID, the default value will be an ID from the appropriate database table.

In the example above, the state type `open` has the `id=4` in the `ticket_state` table.

Note: The `Default` key must refer to a constant value. Substituting it with search terms, regular expressions or similar constructs is currently not supported.

`Description` Defines the description of the field. The description is displayed next to the label as a bubble icon and shows a tooltip when the mouse is hovered.

`Disabled` Defines whether the field is displayed in a disabled state. A disabled field means that the field is read-only. The field value will, however, be sent by submitting the form.

`Hidden` Defines whether the field will be hidden. A hidden field is still part of the form and its value will be sent by submitting the form, but it will not be displayed in the user interface.

`Hint` Defines explanation text for the field which is displayed below the input element as help text.

`Required` Defines whether the field is mandatory. A mandatory field cannot be empty.

The fields can be grouped into form groups. The form groups can contain other form groups or fields. The form groups can be displayed in one, two or three column layouts.

```
- Label: <Displayed Label>
  Collapsible: 1
  Collapsed: 1
  Fields:
  - <Field>
  - ...
  - <FormGroup>
  - ...
  - ColumnLayout: N # Repeated N-times (2 or 3)
    Fields:
    - <Field>
    - ...
    - <FormGroup>
    - ...
```

The following keys and values can be used for the form groups.

Warning: The keys marked with the * symbol are mandatory. Skipping them might result in broken configuration and/or non-functional features.

`Label` * Defines the label of the form group.

Collapsible Defines whether the form group is collapsible.

Collapsed Defines whether the form group is collapsed by default.

Fields * Lists the fields and form groups that belong to this form group. There is no limitation on how many fields can be added or how many form groups can be nested.

If the form group has no fields and acts only as placeholder, then this key has to be defined as an empty list.

```
- Label: Placeholder Form Group
  Fields: []
```

ColumnLayout Defines the grid in the form component. The fields are displayed in one column layout by default. This key can be used only for the two or three column layouts. The width of the columns are the same, 50%-50% for two column layout and 33%-33%-33% for three column layout.

```
- Label: Two Column Example
  - ColumnLayout: 2
    Fields:
    - Name: IsVisibleForCustomer
  - ColumnLayout: 2
    Fields:
    - Name: MarkAsImportant
```

Business Cards

Business cards are used throughout the system to display additional information about agent users.

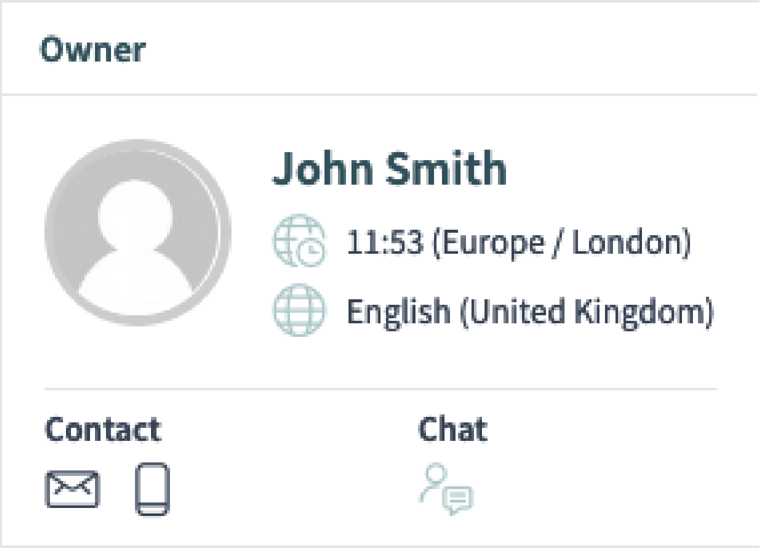


Fig. 56: An Example of the Ticket *Owner* Business Card

The configuration of this business card is located in the AgentFrontend::BusinessCard::User system configuration setting in YAML format.

AdditionalProperties This key is used to show additional user fields in the agent business card. Each field is referenced by its `Name` key. It is possible to customize its `DisplayName` and control visibility via `IsVisible` flag (1 shows it, 0 hides it).

`Contact` This key defines contact options in agent business cards. Each property item is referencing a user field with some contact information via the `Name` key (i.e. `UserEmail`), an icon to show the user (regular weight only!) and can be made visible via the `IsVisible: 1` key. By default, clicking on these contact icons will copy the value of the user field to the clipboard.

Alternatively, the `Link` key can be specified to show a URL instead. Each `Link` key supports the TemplateToolkit syntax for replacements, and will receive all of the field values of the user via the `Data` variable.

`Chat` This key adds an icon to call the chat function directly from a business card.

`Header` This key defines the information shown in the header of a business card. Each field is referenced by its `Name` key. It is only possible to control the visibility via `IsVisible` flag (1 shows it; 0 hides it).

10.8.5 Business Object Reference

This chapter lists all possible values for specific business object configurations. The values can be different in each setting.

Column Names

The columns can be referenced by their name. This section lists the possible column names for specific business object types.

Note: The columns marked with the symbol are sortable. The default limit for the multi-sorting feature is a maximum of three columns at a time, unless explicitly mentioned differently.

`ChatRequest` Possible column names:

- `Action`
- `Channel`
- `CreateTime`
- `Description`
- `RequesterName`
- `RequesterType`
- `Type`

Note: This business object type does not support the multi-sorting feature. Sorting is supported only by a single column at a time.

`CustomerCompany` Possible column names depend on the available fields in the `CustomerCompany###Map` configuration array. For the default database back end, these include:

- `CustomerCompanyCity`
- `CustomerCompanyComment`
- `CustomerCompanyCountry`
- `CustomerCompanyName`

- CustomerCompanyStreet
- CustomerCompanyURL
- CustomerCompanyZIP
- CustomerID
- ValidID

Additional column names which are always available:

- ClosedTickets
- Edit
- OpenTickets

CustomerUser Possible column names depend on the available fields in the CustomerUser###Map and the CustomerCompany###Map configuration arrays. For the default database back ends, these include:

- CustomerCompanyCity
- CustomerCompanyComment
- CustomerCompanyCountry
- CustomerCompanyName
- CustomerCompanyStreet
- CustomerCompanyURL
- CustomerCompanyValidID
- CustomerCompanyZIP
- CustomerID
- UserCity
- UserComment
- UserCountry
- UserCustomerID
- UserEmail
- UserFax
- UserFirstname
- UserLastname
- UserLogin
- UserMobile
- UserPassword
- UserPhone
- UserStreet
- UserTitle
- UserZip
- ValidID

Additional column names which are always available:

- `Chat`
- `ClosedTickets`
- `CreateTicket`
- `Edit`
- `OpenTickets`
- `SwitchToCustomer`

`FormDraft` Possible column names:

- `Delete`
- `Saved`
- `Title`
- `Type`

`KnowledgeBaseArticle` Possible column names:

- `Category`
- `Changed`
- `Created`
- `Language`
- `Number`
- `State`
- `Title`
- `Valid`

`KnowledgeBaseArticleAttachment` Possible column names:

- `ContentType`
- `CreateTime`
- `Download`
- `Filename`
- `Filesize`
- `Preview`

Note: This business object type does not support the multi-sorting feature. Sorting is supported only by a single column at a time.

`Search` Possible column names:

- `Result`
- `Source`
- `Type`

`Statistic` Possible column names:

- `Changed`
- `Created`
- `ObjectName`
- `ObjectType`
- `StatNumber`
- `StatType`
- `Title`
- `Valid`

Note: This business object type does not support the multi-sorting feature. Sorting is supported only by a single column at a time.

`StatisticReport` Possible column names:

- `ChangeTime`
- `CreateTime`
- `CronDefinition`
- `Description`
- `Language`
- `Name`
- `Valid`

Note: This business object type does not support the multi-sorting feature. Sorting is supported only by a single column at a time.

`Ticket` Possible column names:

- `Age`
- `ArticleTree`
- `Changed`
- `Created`
- `CreatedBy`
- `CustomerCompanyName`
- `CustomerID`
- `CustomerName`
- `CustomerUserID`
- `EscalationResponseTime`
- `EscalationSolutionTime`
- `EscalationTime`
- `EscalationUpdateTime`

- `Lock`
- `Owner`
- `PendingTime`
- `Priority`
- `Queue`
- `Responsible`
- `Sender`
- `Service`
- `SLA`
- `State`
- `Subject`
- `TicketNumber`
- `Title`
- `Type`
- `Watch`

`TicketArticle` Possible column names:

- `ArticleNumber`
- `ArticleProperties`
- `Attachment`
- `Channel`
- `CreateTime`
- `Direction`
- `Sender`
- `Status`
- `Subject`

Note: This business object type does not support the multi-sorting feature. Sorting is supported only by a single column at a time.

`TicketAttachment` Possible column names:

- `ContentType`
- `CreateTime`
- `Direction`
- `Download`
- `Filename`
- `Filesize`
- `Preview`

Note: This business object type does not support the multi-sorting feature. Sorting is supported only by a single column at a time.

`WebNotification` Possible column names:

- `CreateTime`
- `Name`
- `ObjectReference`
- `ObjectType`
- `Subject`

Filter Names

Filters can be referenced by their names. This section lists the possible filter names for specific business object types. Each filter has a value type displayed next to it that can be used to quickly jump to the reference.

See also:

The list of possible filter value types can be found in the *Filter Value Types* chapter.

Note: The filters marked with the × (multiple) symbol support array values for multiple matches.

`ChatRequest` The following filter names and values can be used for the *Chat Request* business object lists.

`ChannelIDs`: ***array*** × Defines the IDs of the chat channels.

`Create_DateTimeRange`: ***absolute date*** Defines the absolute range of the chat request creation time.

`Create_DateTimeRelative`: ***relative date*** Defines the relative range of the chat request creation time.

`RequesterType`: ***array*** × Defines the type of the user that initiated the chat request.

`User` The user is an agent.

`Customer` The user is a customer user.

`Public` The user is anonymous (public user).

`Type`: ***array*** × Defines the type of the chat request.

`Invitation` The invitation to a chat initiated by another agent user.

`Request` A chat request initiated by a customer user or a public user.

`CustomerCompany` The following filter names and values can be used for the *Customer* business object lists.

`CustomerCompanyCity`: ***string*** Defines the text when searching for the customer city. Wildcard * can be used. The search is case insensitive.

`CustomerCompanyComment`: ***string*** Defines the text when searching for the customer comment. Wildcard * can be used. The search is case insensitive.

CustomerCompanyCountry: *string* Defines the text when searching for the customer country. Wildcard * can be used. The search is case insensitive.

CustomerCompanyName: *string* Defines the text when searching for the customer name. Wildcard * can be used. The search is case insensitive.

CustomerCompanyStreet: *string* Defines the text when searching for the customer street. Wildcard * can be used. The search is case insensitive.

CustomerCompanyURL: *string* Defines the text when searching for the customer URL. Wildcard * can be used. The search is case insensitive.

CustomerCompanyZIP: *string* Defines the text when searching for the customer postcode. Wildcard * can be used. The search is case insensitive.

CustomerID: *string* Defines the text when searching for the customer ID. Wildcard * can be used. The search is case insensitive.

ValidID: *array* × Defines the IDs of the customer validity. By default, these include:

- `1` = valid
- `2` = invalid
- `3` = invalid-temporarily

CustomerUser The following filter names and values can be used for the *Customer User* business object lists.

CustomerCompany_CustomerCompanyCity: *string* Defines the text when searching for the customer city. Wildcard * can be used. The search is case insensitive.

CustomerCompany_CustomerCompanyComment: *string* Defines the text when searching for the customer comment. Wildcard * can be used. The search is case insensitive.

CustomerCompany_CustomerCompanyCountry: *string* Defines the text when searching for the customer country. Wildcard * can be used. The search is case insensitive.

CustomerCompany_CustomerCompanyName: *string* Defines the text when searching for the customer name. Wildcard * can be used. The search is case insensitive.

CustomerCompany_CustomerCompanyStreet: *string* Defines the text when searching for the customer street. Wildcard * can be used. The search is case insensitive.

CustomerCompany_CustomerCompanyURL: *string* Defines the text when searching for the customer URL. Wildcard * can be used. The search is case insensitive.

CustomerCompany_CustomerCompanyZIP: *string* Defines the text when searching for the customer postcode. Wildcard * can be used. The search is case insensitive.

CustomerCompany_ValidID: *array* × Defines the IDs of the customer validity. By default, these include:

- `1` = valid
- `2` = invalid
- `3` = invalid-temporarily

CustomerID: *string* Defines the text when searching for the related customer ID. Wildcard * can be used. The search is case insensitive.

UserCity: *string* Defines the text when searching for the customer user city. Wildcard * can be used. The search is case insensitive.

`UserComment`: *string* Defines the text when searching for the customer user comment. Wildcard * can be used. The search is case insensitive.

`UserCountry`: *string* Defines the text when searching for the customer user country. Wildcard * can be used. The search is case insensitive.

`UserCustomerID`: *string* Defines the text when searching for the customer user ID. Wildcard * can be used. The search is case insensitive.

`UserEmail`: *string* Defines the text when searching for the customer user email address. Wildcard * can be used. The search is case insensitive.

`UserFax`: *string* Defines the text when searching for the customer user fax number. Wildcard * can be used. The search is case insensitive.

`UserFirstname`: *string* Defines the text when searching for the customer user first name. Wildcard * can be used. The search is case insensitive.

`UserLastname`: *string* Defines the text when searching for the customer user last name. Wildcard * can be used. The search is case insensitive.

`UserLogin`: *string* Defines the text when searching for the customer user login name (username). Wildcard * can be used. The search is case insensitive.

`UserMobile`: *string* Defines the text when searching for the customer user mobile number. Wildcard * can be used. The search is case insensitive.

`UserPhone`: *string* Defines the text when searching for the customer user phone number. Wildcard * can be used. The search is case insensitive.

`UserStreet`: *string* Defines the text when searching for the customer user street. Wildcard * can be used. The search is case insensitive.

`UserTitle`: *string* Defines the text when searching for the customer user title. Wildcard * can be used. The search is case insensitive.

`UserZip`: *string* Defines the text when searching for the customer user postcode. Wildcard * can be used. The search is case insensitive.

`ValidID`: *array* × Defines the IDs of the customer user validity. By default, these include:

- `1` = valid
- `2` = invalid
- `3` = invalid-temporarily

`KnowledgeBaseArticle` The following filter names and values can be used for the *Knowledge Base Article* business object lists.

`Approved`: *boolean* Defines whether the article has been approved.

`CategoryIDs`: *array* × Defines the IDs of the article categories.

`CreatedUserIDs`: *array* × Defines the IDs of the agents who created the article.

`ItemChange_DateTimeRange`: *absolute date* Defines the absolute range of the article change time.

`ItemChange_DateTimeRelative`: *relative date* Defines the relative range of the article change time.

`ItemCreate_DateTimeRange`: *absolute date* Defines the absolute range of the article create time.

`ItemCreate_DateTimeRelative`: *relative date* Defines the relative range of the article create time.

Keyword: *string* Defines the text when searching for the article keyword. Wildcard * can be used. The search is case insensitive.

LanguageIDs: *array* × Defines the IDs of the article language. By default, these include:

- `en`
- `de`

LastChangedUserIDs: *array* × Defines the IDs of the agents who changed the article last.

Number: *string* Defines the text when searching for the article number. Wildcard * can be used.

Rate: *comparison* Compares the article rating in percentage against the supplied value.

StateIDs: *array* × Defines the IDs of the article state. By default, these include:

- `1` = internal (agent)
- `2` = external (customer)
- `3` = public (all)

Title: *string* Defines the text when searching for the article title. Wildcard * can be used. The search is case insensitive.

ValidIDs: *array* × Defines the IDs of the article validity. By default, these include:

- `1` = valid
- `2` = invalid
- `3` = invalid-temporarily

Votes: *comparison* Compares the number of votes for the article against the supplied value.

What: *string* Defines the text when searching through the article full text. Wildcard * can be used. The search is case insensitive.

KnowledgeBaseArticleAttachment The following filter names and values can be used for the *Knowledge Base Article Attachment* business object lists.

Create_DateTimeRange: *absolute date* Defines the absolute range of the attachment create time.

Create_DateTimeRelative: *relative date* Defines the relative range of the attachment create time.

Filename: *string* Defines the text when searching for the attachment filename. Wildcard * and regular expressions can be used. The search is case insensitive.

Type: *string* Defines the text when searching for the attachment MIME type. Wildcard * and regular expressions can be used. The search is case insensitive.

Search The following filter names and values can be used for the *Search* business object lists.

Appointment Calendar: *string* Defines the calendar in which to search for the appointment. Please set it to a string value formatted as:

CalendarID# Where `#` is the ID of the calendar, i.e. `CalendarID1`.

> **Warning:** This filter is applicable only when the filter `DocumentType` is defined and set to the value `Appointment`. Otherwise, it will be ignored.

Appointment Schedule: *string* Defines the type of the appointment.

AllDay The appointment is an all day appointment (i.e. it does not have the time component).

TimeFrame The appointment is a time frame appointment (i.e. it has the time component).

> **Warning:** This filter is applicable only when the filter `DocumentType` is defined and set to the value `Appointment`. Otherwise, it will be ignored.

DocumentType: ***string*** Defines the type of document to be searched. By default, these include:

- `Appointment` = Calendar Appointments
- `FAQ` = Knowledge Base Articles
- `Ticket` = Tickets

FAQ FAQ Categories: ***string*** Defines the knowledge base article category for which you are searching. Please set it to a string value formatted as:

CategoryID# Where `#` is the ID of the category, i.e. `CategoryID1`.

> **Warning:** This filter is applicable only when the filter `DocumentType` is defined and set to the value `FAQ`. Otherwise, it will be ignored.

FAQ FAQ Languages: ***string*** Defines the knowledge base article language for which you are searching. Please set it to a string value formatted as:

LanguageID# Where `#` is the ID of the language, i.e. `LanguageID1`.

> **Warning:** This filter is applicable only when the filter `DocumentType` is defined and set to the value `FAQ`. Otherwise, it will be ignored.

FAQ Has attachments: ***string*** Defines the presence of knowledge base article attachments.

HasAttachments The article has at least one attachment.

> **Warning:** This filter is applicable only when the filter `DocumentType` is defined and set to the value `FAQ`. Otherwise, it will be ignored.

FAQ State: ***string*** Defines the knowledge base article state for which you are searching. Please set it to a string value formatted as:

StateID# Where `#` is the ID of the state, i.e. `StateID1`.

> **Warning:** This filter is applicable only when the filter `DocumentType` is defined and set to the value `FAQ`. Otherwise, it will be ignored.

Ticket Has attachments: ***string*** Defines the presence of the ticket article attachments.

HasAttachments The ticket article has at least one attachment.

> **Warning:** This filter is applicable only when the filter `DocumentType` is defined and set to the value `Ticket`. Otherwise, it will be ignored.

`Ticket Owner`: ***string*** Defines the ID of the agent who is the ticket owner. Please set it to a string value formatted as:

`OwnerID#` Where `#` is the ID of the agent, i.e. `OwnerID1`.

> **Warning:** This filter is applicable only when the filter `DocumentType` is defined and set to the value `Ticket`. Otherwise, it will be ignored.

`Ticket Priority`: ***string*** Defines the ID of the ticket priority. Please set it to a string value formatted as:

`PriorityID#` Where `#` is the ID of the priority, i.e. `PriorityID1`.

> **Warning:** This filter is applicable only when the filter `DocumentType` is defined and set to the value `Ticket`. Otherwise, it will be ignored.

`Ticket Queue`: ***string*** Defines the ID of the ticket queue. Please set it to a string value formatted as:

`QueueID#` Where `#` is the ID of the queue, i.e. `QueueID1`.

> **Warning:** This filter is applicable only when the filter `DocumentType` is defined and set to the value `Ticket`. Otherwise, it will be ignored.

`Ticket Responsible`: ***string*** Defines the ID of the agent who is the ticket responsible. Please set it to a string value formatted as:

`ResponsibleID#` Where `#` is the ID of the agent, i.e. `ResponsibleID1`.

> **Warning:** This filter is applicable only when the filter `DocumentType` is defined and set to the value `Ticket`. Otherwise, it will be ignored.

`Ticket Sender type`: ***string*** Defines the sender of the ticket articles.

`SenderTypeAgent` The sender of the ticket article is an agent user.

`SenderTypeCustomer` The sender of the ticket article is a customer user.

`SenderTypeSystem` The sender of the ticket article is a system user.

> **Warning:** This filter is applicable only when the filter `DocumentType` is defined and set to the value `Ticket`. Otherwise, it will be ignored.

`Ticket Service Level Agreement (SLA)`: ***string*** Defines the ID of the ticket SLA. Please set it to a string value formatted as:

SLAID# Where # is the ID of the SLA, i.e. `SLAID1`.

> **Warning:** This filter is applicable only when the filter `DocumentType` is defined and set to the value `Ticket`. Otherwise, it will be ignored.

Ticket Service: *string* Defines the ID of the ticket service. Please set it to a string value formatted as:

ServiceID# Where # is the ID of the service, i.e. `ServiceID1`.

> **Warning:** This filter is applicable only when the filter `DocumentType` is defined and set to the value `Ticket`. Otherwise, it will be ignored.

Ticket State: *string* Defines the ID of the ticket state. Please set it to a string value formatted as:

StateID# Where # is the ID of the state, i.e. `StateID1`.

> **Warning:** This filter is applicable only when the filter `DocumentType` is defined and set to the value `Ticket`. Otherwise, it will be ignored.

Ticket Visible to customer: *string* Defines whether the ticket articles are visible to the customer user.

IsVisibleForCustomer Ticket article is visible to the customer user.

> **Warning:** This filter is applicable only when the filter `DocumentType` is defined and set to the value `Ticket`. Otherwise, it will be ignored.

Statistic The following filter names and values can be used for the *Statistic* business object lists.

StatNumber: *string* Defines the text when searching for the statistic number. Wildcard * can be used.

StatType: *array* × Defines the statistic type for which you are searching.

static The statistic is of the static type.

dynamic The statistic is of the dynamic type.

Title: *string* Defines the text when searching for the statistic name. Wildcard * can be used. The search is case insensitive.

ObjectName: *array* × Defines the statistic object name to be searched. By default, these include:

- `Ticketlist`
- `TicketAccountedTime`
- `TicketAccumulation`
- `TicketSolutionResponseTime`
- `FAQAccess`
- `StateAction`

ObjectType: *array* × Defines the statistic object type to be searched.

DynamicList The object type is a list.

DynamicMatrix The object type is a matrix.

Static The object type is static.

Valid: *array* × Defines the IDs of the statistic validity. By default, these include:

- `1` = valid
- `0` = invalid

StatisticReport The following filter names and values can be used for the *Report* business object lists.

Name: *string* Defines the text when searching for the report name. Wildcard * can be used. The search is case insensitive.

Description: *string* Defines the text when searching for the report description. Wildcard * can be used. The search is case insensitive.

LanguageID: *string* Defines the report language code, i.e. `de`.

ValidID: *array* × Defines the IDs of the report validity. By default, these include:

- `1` = valid
- `2` = invalid
- `3` = invalid-temporarily

Ticket The following filter names and values can be used for the *Ticket* business object lists.

AgentCreator: *boolean* Defines whether the ticket was created by the current agent.

AgentInvolved: *boolean* Defines whether the current agent is involved in the ticket.

AgentOwner: *boolean* Defines whether the current agent is the ticket owner.

AgentQueues: *boolean* Defines whether the ticket is in one of the subscribed queues of the current agent. Agents can subscribe to queues via their *My Queues* personal preference.

AgentResponsible: *boolean* Defines whether the current agent is the ticket responsible.

AgentServices: *boolean* Defines whether the ticket has one of the subscribed services of the current agent. Agents can subscribe to services via their *My Services* personal preference.

AgentTicketFlagSeen: *string* Defines whether the current agent has read the ticket.

Unread All articles are unread by the current agent.

Read All articles are read by the current agent.

AgentWatcher: *boolean* Defines whether the current agent is watching the ticket.

ArticleCreate_DateTimeRange: *absolute date* Defines the absolute range of the ticket article creation time.

ArticleCreate_DateTimeRelative: *relative date* Defines the relative range of the ticket article creation time.

Chat_ChatterName: *string* Defines the text when searching for the chat article participants. Wildcard * can be used. The search is case insensitive.

Chat_MessageText: *string* Defines the text when searching for the chat article message. Wildcard * can be used. The search is case insensitive.

CreatedQueueIDs: *array* × Defines the IDs of the queues where the ticket was originally created.

CreatedUserIDs: *array* × Defines the IDs of the agents who created the ticket.

CustomerID: *string* Defines the text when searching for the ticket customer ID field. Use this filter for a complex search. Wildcard * can be used. The search is case insensitive.

CustomerIDRaw: *string* Defines the text when searching for the ticket customer ID field. Use this filter for an exact match. The search is case insensitive.

CustomerUserID: *string* Defines whether the ticket is accessible to a specific customer user in the external interface. Supports only exact match. The search is case insensitive.

CustomerUserLogin: *string* Defines the text when searching for the ticket customer user ID field. Use this filter for the complex search. Wildcard * can be used. The search is case insensitive.

CustomerUserLoginRaw: *string* Defines the text when searching for the ticket customer user ID field. Use this filter for an exact match. The search is case insensitive.

Fulltext: *string* Defines the text when searching for the ticket article full text. Wildcard * can be used. The search is case insensitive.

LockIDs: *array* × Defines the IDs of the ticket lock types. By default, these include:

- `1` = unlock
- `2` = lock
- `3` = tmp_lock

MIMEBase_AttachmentName: *string* Defines the text when searching for the article attachment names. Wildcard * can be used. The search is case insensitive.

MIMEBase_Bcc: *string* Defines the text when searching for the article blind carbon copy field. Wildcard * can be used. The search is case insensitive.

MIMEBase_Body: *string* Defines the text when searching for the article body. Wildcard * can be used. The search is case insensitive.

MIMEBase_Cc: *string* Defines the text when searching for the article carbon copy field. Wildcard * can be used. The search is case insensitive.

MIMEBase_From: *string* Defines the text when searching for the article sender field. Wildcard * can be used. The search is case insensitive.

MIMEBase_Subject: *string* Defines the text when searching for the article subject field. Wildcard * can be used. The search is case insensitive.

MIMEBase_To: *string* Defines the text when searching for the article recipient field. Wildcard * can be used. The search is case insensitive.

OwnerIDs: *array* × Defines the IDs of the agents who are the ticket owners.

PriorityIDs: *array* × Defines the IDs of the ticket priorities.

QueueIDs: *array* × Defines the IDs of the ticket queues.

ResponsibleIDs: *array* × Defines the IDs of the agents who are the ticket responsibles.

SearchInArchive: *string* Defines how the search behaves related to the ticket archive status.

ArchivedTickets Search in archived tickets only.

NotArchivedTickets Search in unarchived tickets only.

AllTickets Search in all tickets.

ServiceIDs: *array* × Defines the IDs of the ticket services.

SLAIDs: *array* × Defines the IDs of the ticket SLAs.

SMS_PhoneNumbers: *string* Defines the text when searching for the SMS article recipient phone numbers. Wildcard * can be used. The search is case insensitive.

SMS_Text: *string* Defines the text when searching for the SMS article body. Wildcard * can be used. The search is case insensitive.

SMS_TransactionNumbers: *string* Defines the text when searching for the SMS article transaction numbers. Wildcard * can be used. The search is case insensitive.

StateIDs: *array* × Defines the IDs of the ticket states.

StateType: *string* Defines the ticket state category.

Open Search in open tickets only.

Closed Search in closed tickets only.

> **Warning:** Despite its name, this filter does not apply to the actual *ticket state types*. `Open` and `Closed` are not valid state types. They are group constructs included for compatibility reasons. If a ticket is in a state which has any of the state types defined in the Ticket::ViewableStateType setting, it will be considered as open; otherwise, it is closed.

StateTypeIDs: *array* × Defines the IDs of the ticket state types.

TicketChange_DateTimeRange: *absolute date* Defines the absolute range of the ticket change time.

TicketChange_DateTimeRelative: *relative date* Defines the relative range of the ticket change time.

TicketClose_DateTimeRange: *absolute date* Defines the absolute range of the ticket close time.

TicketClose_DateTimeRelative: *relative date* Defines the relative range of the ticket close time.

TicketCreate_DateTimeRange: *absolute date* Defines the absolute range of the ticket create time.

TicketCreate_DateTimeRelative: *relative date* Defines the relative range of the ticket create time.

TicketEscalation_DateTimeRange: *absolute date* Defines the absolute range of the ticket escalation time.

TicketEscalation_DateTimeRelative: *relative date* Defines the relative range of the ticket escalation time.

TicketLastChange_DateTimeRange: *absolute date* Defines the absolute range of the ticket last change time.

TicketLastChange_DateTimeRelative: *relative date* Defines the relative range of the ticket last change time.

TicketNumber: *string* Defines the text when searching for the ticket number. Wildcard * can be used.

TicketPending_DateTimeRange: *absolute date* Defines the absolute range of the ticket pending time.

`TicketPending_DateTimeRelative`: *relative date* Defines the relative range of the ticket pending time.

`Title`: *string* Defines the text when searching for the ticket title. Wildcard * can be used. The search is case insensitive.

`TypeIDs`: *array* × Defines the text when searching for the ticket type.

`WatchUserIDs`: *array* × Defines the IDs of the agents who are watching the ticket.

`TicketArticle` The following filter names and values can be used for the *Ticket Article* business object lists.

`CommunicationChannelID`: *array*

`IsVisibleForCustomer`: *string*

`SenderTypeID`: *array*

`TicketAttachment` The following filter names and values can be used for the *Ticket Attachment* business object lists.

`Article`: *string*

`Create_DateTimeRange`: *absolute date*

`Create_DateTimeRelative`: *relative date*

`Direction`: *array* ×

`Filename`: *string*

`Type`: *string*

`WebNotification` The following filter names and values can be used for the *Web Notification* business object lists.

`Name`: *array* ×

`Subject`: *string*

`ObjectTypes`: *array* ×

`Seen`: *array* ×

`ObjectReferences`: *array* ×

Filter Value Types

Every filter supports exactly one kind of a value type. Depending on this type, the following structures should be used to define it in the YAML configuration.

Boolean A simple value that is considered either true or false.

```
<FilterName1>:
  Value: 1
<FilterName2>:
  Value: 0
```

1 The filter is active (turned on).

0 The filter is inactive (turned off).

String A text value, normally used when searching. Check specific filters for more details.

```
<FilterName1>:
  Value: simple
<FilterName2>:
  Value: 'String with spaces or special characters like @$#!+*'
```

Array An array value normally used to search for multiple values at the same time.

```
<FilterName1>:
  Value:
  - 1
  - 2
  - 3
<FilterName2>:
  Value:
  - 4
  - 5
```

Absolute date The time range value between two absolute dates.

```
<FilterName1>:
  Value:
    Start: '2020-01-01 00:00:00'
    End: '2020-02-01 00:00:00'
<FilterName2>:
  Value:
    Start: '2020-02-01 00:00:00'
    End: '2020-03-01 00:00:00'
```

`Start` Absolute timestamp in ISO format for the start of the range (i.e. `2020-01-01 00:00:00`). The value must be supplied in the current system time zone (configurable via the OTRSTimeZone system setting).

`End` Absolute timestamp in ISO format for the end of the range (i.e. `2020-02-01 00:00:00`). The value must be supplied in the current system time zone (configurable via the OTRSTimeZone system setting).

Relative date The time range value relative to the current time.

```
<FilterName1>:
  Value:
    Start: Last
    Point: 1
    Format: month
<FilterName2>:
  Value:
    Start: Before
    Point: 3
    Format: day
```

`Start` The type of the relative range.

Possible values for the `Start` key:

- `Last` = within the last…
- `Before` = more than …ago

`Point` Integer value of the time unit (i.e. `3`).

`Format` Time unit of the relative range.

Possible values for the `Format` key:

```
minute
hour
day
week
month
year
```

Comparison The comparison with the supplied value.

```
<FilterName1>:
  Value:
    Type: Equals
    Value: 25
<FilterName2>:
  Value:
    Start: GreaterThan
    Value: 50
```

`Type` The type of the comparison.

Possible values for the `Type` key:

- `Equals` = equals …
- `GreaterThan` = greater than …
- `GreaterThanEquals` = greater than or equals …
- `SmallerThan` = smaller than …
- `SmallerThanEquals` = smaller than or equals …

`Value` Integer value to compare against (i.e. `25`).

Form Fields

The form fields can be defined via their `Name` key. This section lists the values of the `Name` key for the specific settings.

See also:

Form fields can be made mandatory by setting their *Required* key to `1`.

Possible field names for ticket actions:

`Forms###AgentFrontend::Ticket::Action::Close` Configurable form for the *Close Ticket* action.

Link to the reference of the system configuration:

- Forms###AgentFrontend::Ticket::Action::Close

Possible values for the `Name` key of the form field:

```
AccountedTime
AddMessage
Attachments
Bcc
Body
Cc
CustomerID
CustomerUserID
DynamicField
HistoryComment
HistoryType
InformAgent
InvolvedAgent
IsVisibleForCustomer
MarkAsImportant
Messages
OwnerID
PriorityID
QueueID
RelevantKnowledge
ResponsibleID
SenderType
ServiceID
Signature
SLAID
StandardTemplateID
StateID
Subject
Title
To
TypeID
```

`Forms###AgentFrontend::Ticket::Action::Customer` Configurable form for the *Change Customer* action.

Link to the reference of the system configuration:

- Forms###AgentFrontend::Ticket::Action::Customer

Possible values for the `Name` key of the form field:

```
CustomerUserID
CustomerID
```

`Forms###AgentFrontend::Ticket::Action::EmailOutbound` Configurable form for the *Send Email Outbound* action.

Link to the reference of the system configuration:

- Forms###AgentFrontend::Ticket::Action::EmailOutbound

Possible values for the `Name` key of the form field:

```
AccountedTime
Attachments
Bcc
```

(continues on next page)

(continued from previous page)

```
Body
Cc
CustomerID
CustomerUserID
DynamicField
EmailSecurity
From
HistoryComment
HistoryType
InformAgent
InvolvedAgent
IsVisibleForCustomer
MarkAsImportant
Messages
OwnerID
PriorityID
QueueID
RelevantKnowledge
ResponsibleID
SenderType
ServiceID
Signature
SLAID
StandardTemplateID
StateID
Subject
Title
To
TypeID
```

`Forms###AgentFrontend::Ticket::Action::FreeText` Configurable form for the *Change Free Fields* action.

Link to the reference of the system configuration:

- Forms###AgentFrontend::Ticket::Action::FreeText

Possible values for the `Name` key of the form field:

```
AccountedTime
AddMessage
Attachments
Bcc
Body
Cc
CustomerID
CustomerUserID
DynamicField
HistoryComment
HistoryType
InformAgent
InvolvedAgent
IsVisibleForCustomer
MarkAsImportant
```

(continues on next page)

(continued from previous page)

```
Messages
OwnerID
PriorityID
QueueID
RelevantKnowledge
ResponsibleID
SenderType
ServiceID
Signature
SLAID
StandardTemplateID
StateID
Subject
Title
To
TypeID
```

`Forms###AgentFrontend::Ticket::Action::Merge` Configurable form for the *Merge Ticket* action.

Link to the reference of the system configuration:

- Forms###AgentFrontend::Ticket::Action::Merge

Possible values for the `Name` key of the form field:

```
AddMessage
Attachments
Body
DynamicField
From
HistoryComment
HistoryType
IsVisibleForCustomer
MarkAsImportant
Messages
RelevantKnowledge
SenderType
Subject
To
```

`Forms###AgentFrontend::Ticket::Action::Move` Configurable form for the *Move Ticket* action.

Link to the reference of the system configuration:

- Forms###AgentFrontend::Ticket::Action::Move

Possible values for the `Name` key of the form field:

```
AccountedTime
AddMessage
Attachments
Bcc
Body
Cc
CustomerID
```

(continues on next page)

(continued from previous page)

```
CustomerUserID
DynamicField
HistoryComment
HistoryType
InformAgent
InvolvedAgent
IsVisibleForCustomer
MarkAsImportant
Messages
OwnerID
PriorityID
QueueID
RelevantKnowledge
ResponsibleID
SenderType
ServiceID
Signature
SLAID
StandardTemplateID
StateID
Subject
Title
To
TypeID
```

`Forms###AgentFrontend::Ticket::Action::Note` Configurable form for the *Add Note* action.

Link to the reference of the system configuration:

- Forms###AgentFrontend::Ticket::Action::Note

Possible values for the `Name` key of the form field:

```
AccountedTime
Attachments
Bcc
Body
Cc
CustomerID
CustomerUserID
DynamicField
HistoryComment
HistoryType
InformAgent
InvolvedAgent
IsVisibleForCustomer
MarkAsImportant
Messages
OwnerID
PriorityID
QueueID
RelevantKnowledge
ResponsibleID
SenderType
```

(continues on next page)

(continued from previous page)

```
ServiceID
Signature
SLAID
StandardTemplateID
StateID
Subject
Title
To
TypeID
```

`Forms###AgentFrontend::Ticket::Action::Owner` Configurable form for the *Change Owner* action.

Link to the reference of the system configuration:

- Forms###AgentFrontend::Ticket::Action::Owner

Possible values for the `Name` key of the form field:

```
AccountedTime
AddMessage
Attachments
Bcc
Body
Cc
CustomerID
CustomerUserID
DynamicField
HistoryComment
HistoryType
InformAgent
InvolvedAgent
IsVisibleForCustomer
MarkAsImportant
Messages
OwnerID
PriorityID
QueueID
RelevantKnowledge
ResponsibleID
SenderType
ServiceID
Signature
SLAID
StandardTemplateID
StateID
Subject
Title
To
TypeID
```

`Forms###AgentFrontend::Ticket::Action::Pending` Configurable form for the *Set Pending Time* action.

Link to the reference of the system configuration:

- Forms###AgentFrontend::Ticket::Action::Pending

Possible values for the `Name` key of the form field:

```
AccountedTime
AddMessage
Attachments
Bcc
Body
Cc
CustomerID
CustomerUserID
DynamicField
HistoryComment
HistoryType
InformAgent
InvolvedAgent
IsVisibleForCustomer
MarkAsImportant
Messages
OwnerID
PriorityID
QueueID
RelevantKnowledge
ResponsibleID
SenderType
ServiceID
Signature
SLAID
StandardTemplateID
StateID
Subject
Title
To
TypeID
```

`Forms###AgentFrontend::Ticket::Action::PhoneCallInbound` Configurable form for the *Add Phone Call Inbound* action.

Link to the reference of the system configuration:

- Forms###AgentFrontend::Ticket::Action::PhoneCallInbound

Possible values for the `Name` key of the form field:

```
AccountedTime
Attachments
Bcc
Body
Cc
CustomerID
CustomerUserID
DynamicField
HistoryComment
HistoryType
InformAgent
```

(continues on next page)

(continued from previous page)

```
InvolvedAgent
IsVisibleForCustomer
MarkAsImportant
Messages
OwnerID
PriorityID
QueueID
RelevantKnowledge
ResponsibleID
SenderType
ServiceID
Signature
SLAID
StandardTemplateID
StateID
Subject
Title
To
TypeID
```

`Forms###AgentFrontend::Ticket::Action::PhoneCallOutbound` Configurable form for the *Add Phone Call Outbound* action.

Link to the reference of the system configuration:

- Forms###AgentFrontend::Ticket::Action::PhoneCallOutbound

Possible values for the `Name` key of the form field:

```
AccountedTime
Attachments
Bcc
Body
Cc
CustomerID
CustomerUserID
DynamicField
HistoryComment
HistoryType
InformAgent
InvolvedAgent
IsVisibleForCustomer
MarkAsImportant
Messages
OwnerID
PriorityID
QueueID
RelevantKnowledge
ResponsibleID
SenderType
ServiceID
Signature
SLAID
StandardTemplateID
```

(continues on next page)

(continued from previous page)

```
StateID
Subject
Title
To
TypeID
```

`Forms###AgentFrontend::Ticket::Action::Priority` Configurable form for the *Change Priority* action.

Link to the reference of the system configuration:

- Forms###AgentFrontend::Ticket::Action::Priority

Possible values for the `Name` key of the form field:

```
AccountedTime
AddMessage
Attachments
Bcc
Body
Cc
CustomerID
CustomerUserID
DynamicField
HistoryComment
HistoryType
InformAgent
InvolvedAgent
IsVisibleForCustomer
MarkAsImportant
Messages
OwnerID
PriorityID
QueueID
RelevantKnowledge
ResponsibleID
SenderType
ServiceID
Signature
SLAID
StandardTemplateID
StateID
Subject
Title
To
TypeID
```

`Forms###AgentFrontend::Ticket::Action::Responsible` Configurable form for the *Change Responsible* action.

Link to the reference of the system configuration:

- Forms###AgentFrontend::Ticket::Action::Responsible

Possible values for the `Name` key of the form field:

```
AccountedTime
AddMessage
Attachments
Bcc
Body
Cc
CustomerID
CustomerUserID
DynamicField
HistoryComment
HistoryType
InformAgent
InvolvedAgent
IsVisibleForCustomer
MarkAsImportant
Messages
OwnerID
PriorityID
QueueID
RelevantKnowledge
ResponsibleID
SenderType
ServiceID
Signature
SLAID
StandardTemplateID
StateID
Subject
Title
To
TypeID
```

`Forms###AgentFrontend::Ticket::Action::SmsOutbound` Configurable form for the *Send SMS Outbound* action.

Link to the reference of the system configuration:

- Forms###AgentFrontend::Ticket::Action::SmsOutbound

Possible values for the `Name` key of the form field:

```
AccountedTime
Body
CustomerID
CustomerUserID
DynamicField
FlashMessage
HistoryComment
HistoryType
InformAgent
InvolvedAgent
IsVisibleForCustomer
MarkAsImportant
Messages
```

(continues on next page)

(continued from previous page)

```
OwnerID
PriorityID
QueueID
RelevantKnowledge
ResponsibleID
SenderType
ServiceID
Signature
SLAID
StandardTemplateID
StateID
Title
To
TypeID
```

Possible field names for article actions:

`Forms###AgentFrontend::TicketArticle::Action::Forward` Configurable form for the *Forward via Email* action.

Link to the reference of the system configuration:

- Forms###AgentFrontend::TicketArticle::Action::Forward

Possible values for the `Name` key of the form field:

```
AccountedTime
Attachments
Bcc
Body
Cc
DynamicField
EmailSecurity
From
HistoryComment
HistoryType
IsVisibleForCustomer
MarkAsImportant
Messages
RelevantKnowledge
SenderType
StandardTemplateID
StateID
Subject
To
```

`Forms###AgentFrontend::TicketArticle::Action::Redirect` Configurable form for the *Redirect via Email* action.

Link to the reference of the system configuration:

- Forms###AgentFrontend::TicketArticle::Action::Redirect

Possible values for the `Name` key of the form field:

```
AddMessage
Body
From
HistoryType
Messages
RedirectTo
StateID
Subject
To
```

`Forms###AgentFrontend::TicketArticle::Action::Reply` Configurable form for the *Reply via Email* action.

Link to the reference of the system configuration:

- Forms###AgentFrontend::TicketArticle::Action::Reply

Possible values for the `Name` key of the form field:

```
AccountedTime
Attachments
Bcc
Body
Cc
DynamicField
EmailSecurity
From
HistoryComment
HistoryType
IsVisibleForCustomer
MarkAsImportant
Messages
RelevantKnowledge
SenderType
StandardTemplateID
StateID
Subject
To
```

`Forms###AgentFrontend::TicketArticle::Action::ReplyAll` Configurable form for the *Reply to All via Email* action.

Link to the reference of the system configuration:

- Forms###AgentFrontend::TicketArticle::Action::ReplyAll

Possible values for the `Name` key of the form field:

```
AccountedTime
Attachments
Bcc
Body
Cc
DynamicField
EmailSecurity
From
```

(continues on next page)

(continued from previous page)

```
HistoryComment
HistoryType
IsVisibleForCustomer
MarkAsImportant
Messages
RelevantKnowledge
SenderType
StandardTemplateID
StateID
Subject
To
```

Forms###AgentFrontend::TicketArticle::Action::ReplyToNote Configurable form for the *Reply via Note* action.

Link to the reference of the system configuration:

- Forms###AgentFrontend::TicketArticle::Action::ReplyToNote

Possible values for the `Name` key of the form field:

```
AccountedTime
Attachments
AutoInvolvedAgents
Bcc
Body
Cc
CustomerID
CustomerUserID
DynamicField
HistoryComment
HistoryType
InformAgent
InvolvedAgent
IsVisibleForCustomer
MarkAsImportant
Messages
OwnerID
PriorityID
QueueID
RelevantKnowledge
ResponsibleID
SenderType
ServiceID
Signature
SLAID
StandardTemplateID
StateID
Subject
Title
To
TypeID
```

Forms###AgentFrontend::TicketArticle::Action::ReplyViaSms Configurable form for the

Reply via SMS action.

Link to the reference of the system configuration:

- Forms###AgentFrontend::TicketArticle::Action::ReplyViaSms

Possible values for the `Name` key of the form field:

```
AccountedTime
Body
CustomerID
CustomerUserID
DynamicField
FlashMessage
HistoryComment
HistoryType
InformAgent
InvolvedAgent
IsVisibleForCustomer
MarkAsImportant
Messages
OwnerID
PriorityID
QueueID
RelevantKnowledge
ResponsibleID
SenderType
ServiceID
Signature
SLAID
StandardTemplateID
StateID
Title
To
TypeID
```

`Forms###AgentFrontend::TicketArticle::Action::Split` Configurable form for the *Split Article* action.

Link to the reference of the system configuration:

- Forms###AgentFrontend::TicketArticle::Action::Split

Possible values for the `Name` key of the form field:

```
LinkAs
Messages
ProcessID
Target
```

Possible field names for the ticket create properties forms:

`Forms###AgentFrontend::TicketCreate::Email::CreateProperties` Configurable form for the *Properties* widget of the *New Email Ticket* screen.

Link to the reference of the system configuration:

- Forms###AgentFrontend::TicketCreate::Email::CreateProperties

Possible values for the `Name` key of the form field:

```
AccountedTime
Attachments
Bcc
Body
Cc
CustomerID
CustomerUserID
DynamicField
EmailSecurity
HistoryComment
HistoryType
IsVisibleForCustomer
LinkTicketID
LinkType
OwnerID
PriorityID
QueueID
RelevantKnowledge
ResponsibleID
SenderType
ServiceID
Signature
SLAID
StandardTemplateID
StateID
Subject
To
TypeID
```

`Forms###AgentFrontend::TicketCreate::Phone::CreateProperties` Configurable form for the *Properties* widget of the *New Phone Ticket* screen.

Link to the reference of the system configuration:

- Forms###AgentFrontend::TicketCreate::Phone::CreateProperties

Possible values for the `Name` key of the form field:

```
AccountedTime
Attachments
Body
CustomerID
CustomerUserID
DynamicField
From
HistoryComment
HistoryType
IsVisibleForCustomer
LinkTicketID
LinkType
OwnerID
PendingDate
PriorityID
```

(continues on next page)

(continued from previous page)

```
QueueID
RelevantKnowledge
ResponsibleID
SenderType
ServiceID
SLAID
StandardTemplateID
StateID
Subject
To
TypeID
```

`Forms###AgentFrontend::TicketCreate::SMS::CreateProperties` Configurable form for the *Properties* widget of the *New SMS Ticket* screen.

Link to the reference of the system configuration:

- Forms###AgentFrontend::TicketCreate::SMS::CreateProperties

Possible values for the `Name` key of the form field:

```
AccountedTime
Body
CustomerID
CustomerUserID
DynamicField
FlashMessage
HistoryComment
HistoryType
IsVisibleForCustomer
OwnerID
PendingDate
PriorityID
QueueID
RelevantKnowledge
ResponsibleID
Sender
SenderType
ServiceID
SLAID
StandardTemplateID
StateID
Subject
To
TypeID
```

Possible field names for knowledge base article actions:

`Forms###AgentFrontend::KnowledgeBaseArticleCreate::Properties` Configurable form for the *Properties* widget of the *Create Knowledge Base Article* screen.

Link to the reference of the system configuration:

- Forms###AgentFrontend::KnowledgeBaseArticleCreate::Properties

Possible values for the `Name` key of the form field:

```
Approved
Attachments
Category
DynamicFields
Field1
Field2
Field3
Field4
Field5
Field6
Keywords
Language
State
Title
Valid
```

`Forms###AgentFrontend::KnowledgeBaseArticleUpdate::Properties` Configurable form for the *Properties* widget of the *Edit Knowledge Base Article* screen.

Link to the reference of the system configuration:

- Forms###AgentFrontend::KnowledgeBaseArticleUpdate::Properties

Possible values for the `Name` key of the form field:

```
Approved
Attachments
Category
DynamicFields
Field1
Field2
Field3
Field4
Field5
Field6
Keywords
Language
State
Title
Valid
```

Possible field names for customer actions:

`Forms###AgentFrontend::CustomerCompanyCreate::Properties` Configurable form for the *Properties* widget of the *Create Customer* screen.

Link to the reference of the system configuration:

- Forms###AgentFrontend::CustomerCompanyCreate::Properties

Possible values for the `Name` key depend on the available fields in the `CustomerCompany###Map` configuration array. For the default database back end, these include:

```
CustomerCompanyCity
CustomerCompanyComment
CustomerCompanyCountry
```

(continues on next page)

(continued from previous page)

```
CustomerCompanyName
CustomerCompanyStreet
CustomerCompanyURL
CustomerCompanyZIP
CustomerID
ValidID
```

Additional column names which are always available:

```
DataSource
```

> **Warning:** In case *Multiple Customer User Back Ends* are configured, all possible fields must be present in the form configuration in order to be able to modify them. Both configurations must be kept in sync, otherwise it will not be possible to modify the customer records.

`Forms###AgentFrontend::CustomerCompanyUpdate::Properties` Configurable form for the *Properties* widget of the *Edit Customer* screen.

Link to the reference of the system configuration:

- Forms###AgentFrontend::CustomerCompanyUpdate::Properties

Possible values for the `Name` key depend on the available fields in the `CustomerCompany###Map` configuration array. For the default database back end, these include:

```
CustomerCompanyCity
CustomerCompanyComment
CustomerCompanyCountry
CustomerCompanyName
CustomerCompanyStreet
CustomerCompanyURL
CustomerCompanyZIP
CustomerID
ValidID
```

> **Warning:** In case *Multiple Customer User Back Ends* are configured, all possible fields must be present in the form configuration in order to be able to modify them. Both configurations must be kept in sync, otherwise it will not be possible to modify the customer records.

Possible field names for customer user actions:

`Forms###AgentFrontend::CustomerUserCreate::Properties` Configurable form for the *Properties* widget of the *Create Customer User* screen.

Link to the reference of the system configuration:

- Forms###AgentFrontend::CustomerUserCreate::Properties

Possible values for the `Name` key depend on the *Customer User Back Ends* configuration. For the default database back end, these include:

```
UserCity
UserComment
UserCountry
UserCustomerID
UserEmail
UserFax
UserFirstname
UserLastname
UserLogin
UserMobile
UserPassword
UserPhone
UserStreet
UserTitle
UserZip
ValidID
```

For most customer user preference modules registered under `CustomerPersonalPreference###*` namespace, it is also possible to show dedicated form fields. By default, these include:

```
Preference_LoginForbidden
Preference_PGP
Preference_SMIME
Preference_TwoFactor
```

Additional column names which are always available:

```
DataSource
```

> **Warning:** In case of *Multiple Customer User Back Ends*, all possible fields must be present in the form configuration in order to be able to modify them. Both configurations must be kept in sync, otherwise it will not be possible to modify the customer user records.

`Forms###AgentFrontend::CustomerUserUpdate::Properties` Configurable form for the *Properties* widget of the *Edit Customer User* screen.

Link to the reference of the system configuration:

- Forms###AgentFrontend::CustomerUserUpdate::Properties

Possible values for the `Name` key depend on the *Customer User Back Ends* configuration. For the default database back end, these include:

```
UserCity
UserComment
UserCountry
UserCustomerID
UserEmail
UserFax
UserFirstname
UserLastname
UserLogin
UserMobile
```

(continues on next page)

(continued from previous page)

```
UserPassword
UserPhone
UserStreet
UserTitle
UserZip
ValidID
```

For most customer user preference modules registered under `CustomerPersonalPreference###*` namespace, it is also possible to show dedicated form fields. By default, these include:

```
Preference_LoginForbidden
Preference_PGP
Preference_SMIME
Preference_TwoFactor
```

> **Warning:** In case of *Multiple Customer User Back Ends*, all possible fields must be present in the form configuration in order to be able to modify them. Both configurations must be kept in sync, otherwise it will not be possible to modify the customer user records.

Possible field names for calendar appointment actions:

`Forms###AgentFrontend::Calendar::AppointmentCreate::Properties` Configurable form for the *Add Appointment* action.

Link to the reference of the system configuration:

- Forms###AgentFrontend::Calendar::AppointmentCreate::Properties

Possible values for the `Name` key of the form field:

```
AllDay
CalendarID
Description
EndTime
Location
Notification
Recurrence
ResourceID
StartTime
TeamID
TicketPlugin
Title
```

`Forms###AgentFrontend::Calendar::AppointmentUpdate::Properties` Configurable form for the *Edit Appointment* action.

Link to the reference of the system configuration:

- Forms###AgentFrontend::Calendar::AppointmentUpdate::Properties

Possible values for the `Name` key of the form field:

```
AllDay
CalendarID
```

(continues on next page)

(continued from previous page)

```
Description
EndTime
Location
Notification
Recurrence
ResourceID
StartTime
TeamID
TicketPlugin
Title
```

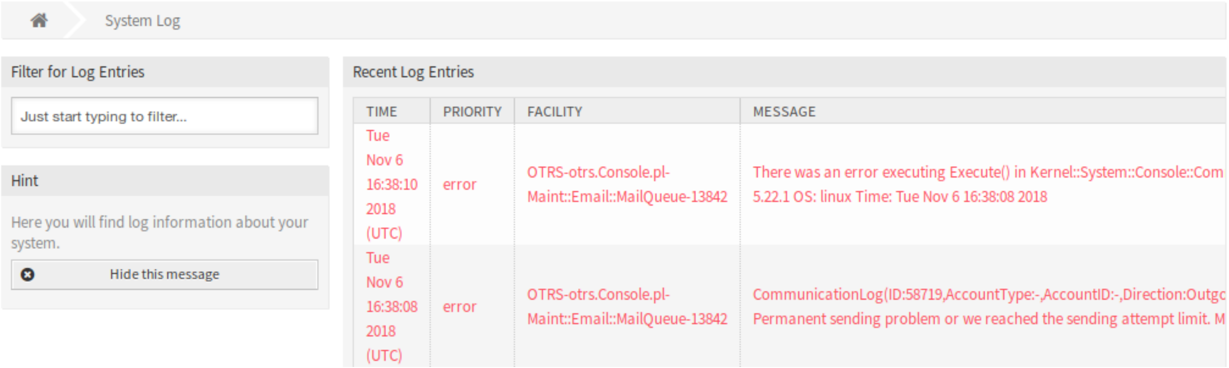
10.9 System Log

Professional systems log their activities in one or more log files to help administrators when troubleshooting or to get an overview of what is going on in their system on a detailed level.

These logs are usually not available to application administrators without a certain level of permissions, and skills, on the operating system.

OTRS allows application administrators to access the system log comfortably by using the graphical interface without the need to have access to the server' s command shell. The administrator can decide which level of logging is needed, to make sure that the log files are not unnecessarily filled.

Use this screen to view log entries of OTRS. The log overview screen is available in the *System Log* module of the *Administration* group.



System Log
Filter for Log Entries
Just start typing to filter...
Hint
Here you will find log information about your system.
Hide this message
Recent Log Entries

TIME	PRIORITY	FACILITY	MESSAGE
Tue Nov 6 16:38:10 2018 (UTC)	error	OTRS-otrs.Console.pl-Maint::Email::MailQueue-13842	There was an error executing Execute() in Kernel::System::Console::Com 5.22.1 OS: linux Time: Tue Nov 6 16:38:08 2018
Tue Nov 6 16:38:08 2018 (UTC)	error	OTRS-otrs.Console.pl-Maint::Email::MailQueue-13842	CommunicationLog(ID:58719,AccountType:-,AccountID:-,Direction:Outgc Permanent sending problem or we reached the sending attempt limit. M

Fig. 57: System Log Screen

Each line in the log contains a timestamp, the log priority, the system component and the log entry itself.

Note: If several log entries are displayed in the log, use the filter box to find a particular log entry by just typing the name to filter.

To adjust the logging settings of the system:

1. Go to *System Configuration* screen.
2. Navigate to *Core* → *Log* in the navigation tree.
3. Review the settings.

CHAPTER 11

Tutorials

Any system requires configuration. Configuring a system should be an easy task and the tools for configuration fit-for-purpose.

OTRS offers several administration tools to configure, monitor, control and extend OTRS.

11.1 Archiving Tickets

OTRS can be used as an audit-proof system. In this case deleting closed tickets may not be a good idea. Therefore a feature has been implemented that allows you to archive tickets.

Tickets that match certain criteria can be marked as *archived*. These tickets are not accessed if an agent do a regular ticket search or run a generic agent job. The system itself does not have to deal with a huge amount of tickets any longer as only the latest tickets are taken into consideration. This can result in a huge performance gain on large systems.

To use the archive feature:

1. Go to the *System Configuration* screen.
2. Search for the setting `Ticket::ArchiveSystem` and enable the setting.
3. Go to the *Generic Agent* screen.
4. Click on the *Add job* button in the left sidebar.
5. Provide a name in the *Job Settings* section, and select proper options to schedule this job.
6. Define the ticket filters in the *Select Tickets* section. It might be a good idea to only archive those tickets in a closed state that have been closed a few months before.
7. In the *Update/Add Ticket Attributes* section, set the field *Archive selected tickets* to *archive tickets*.
8. Click on the *Save* button.
9. Run the job. The system will display all tickets which will be archived when executing the generic agent job.

Note: When you search for tickets, the system default is to search tickets which are not archived. If you want to search through archived tickets also, simply add the *Archive search* filter while defining search criteria.

11.2 Creating New Phone Ticket on Incoming Calls

It is possible to integrate OTRS with third-party computer telephony integration (CTI) solutions and make them work in unison. On as simple action as a click on a button in the CTI incoming call screen, the OTRS *New Phone Ticket* screen can be opened with the customer pre-selected by their phone number.

The only pre-requisite for the example below is that the used CTI software is able to open a link of a certain format in the web browser.

1. First, edit the customer user configuration file and add the phone number field to `CustomerUserSearchFields` (in our case it is aptly named `phone`):

   ```
   CustomerUserSearchFields => [ 'login', 'first_name', 'last_name',
    ↪'customer_id', 'phone' ],
   ```

   Note: For more information on how to provide custom user configuration please see the *Customer User Back Ends* chapter.

   Warning: The customer user configuration is provided as an additional Perl module file. It is your responsibility to check the validity of the supplied file, since any syntax mistakes can lead to a broken system.

   Optionally, you can also add the `phone` field to `CustomerUserListFields` array, if you want to see the telephone number while searching for customer users:

   ```
   CustomerUserListFields => [ 'first_name', 'last_name', 'phone', 'email' ],
   ```

   Note: The customer user configuration mentioned above is taken care of by the *Customer Solution Team* in **OTRS**. Please contact the *Customer Solution Team* via support@otrs.com or in the OTRS Portal.

2. Configure your CTI software to open the following link when you answer the phone:

   ```
   https://otrs.example.com/agent/ticket/create/phone?CustomerUserSearch=1&
    ↪CustomerUserID=callernumber
   https://otrs.example.com/agent/ticket/create/phone?CustomerUserSearch=1&
    ↪CustomerUserID=callerloginname
   ```

   For example, for customer user *John Smith* with login name *johnsmith* and telephone number *12345678*:

```
https://otrs.example.com/agent/ticket/create/phone?CustomerUserSearch=1&
 ↪CustomerUserID=12345678
https://otrs.example.com/agent/ticket/create/phone?CustomerUserSearch=1&
 ↪CustomerUserID=johnsmith
```

Where *otrs.example.com* is the FQDN of the OTRS system in question. Which exact value you use in the end (caller number or login name) depends on what your CTI software can construct the link with.

3. When the ticket create screen is open in OTRS, a search will be conducted for the supplied value. In case only one customer user with that information is found, they will be pre-selected as the ticket customer user, and added to the related fields.

Note: In case you would like to do a search by the customer user email address, first make sure that the `CustomerUserPostMasterSearchFields` key of the customer user configuration is set to an appropriate field. Additionally, you must use an alternative URL query parameter called `CustomerPostMasterSearch=1`, since the email addresses use a different search mechanism. Finally, provide the email address for search as the `CustomerUserID` parameter, but you make sure to URL encode the value because it can contain some unsafe characters.

Note: Mechanism described above works for any of the following ticket create screens:

- New Phone Ticket (default slug: `/agent/ticket/create/phone`)
- New Email Ticket (default slug: `/agent/ticket/create/email`)
- New SMS Ticket (default slug: `/agent/ticket/create/sms`)
- New Process Ticket (default slug: `/agent/ticket/create/process`)

Considering the new process ticket screen consists of a more complex form, you must also provide a correct `ProcessID` parameter in order for the process to be pre-selected. For example:

```
https://otrs.example.com/agent/ticket/create/process?ProcessID=Process-
 ↪9690ae9ae455d8614d570149b8ab1199&CustomerUserSearch=1&
 ↪CustomerUserID=12345678
```

11.3 Custom URL Support

Most OTRS front ends support automation via custom URL addresses, whether for specifying configuration parameters, or automating forms. This mechanism can be used to integrate OTRS easily into existing workflows, or just provide a quick pre-configured overview of objects. There is no limit to number of use-cases.

11.3.1 Triggering Actions

Actions in the agent interface can be automatically triggered via a special URL parameter called `TriggerAction`. The approach outlined below works for any action which is registered for the current screen.

For example, let’s trigger the *Change Free Fields* action available in the ticket detail view.

1. Identify the slug name of the view in question. For the ticket detail view and the ticket with ID **1** this is:

   ```
   /agent/ticket/1
   ```

2. Identify the `BusinessObjectType` of the screen where the action should be triggered. In our case, for the ticket detail view, this is:

   ```
   Ticket
   ```

3. Go to *System Configuration* in the administrator interface and search for the action registration setting in question. In our case this is:

   ```
   AgentFrontend::Ticket::Action###FreeText
   ```

4. Identify the value of the `Component` key in the setting, which is the name of the front end component. In our case this is:

   ```
   TicketFreeText
   ```

5. Construct the action identifier in form of `<BusinessObjectType>::<Component>`. In our case, this is:

   ```
   Ticket::TicketFreeText
   ```

6. Append a query parameter to the URL of the detail view named `TriggerAction`, and set it to the action identifier, which is the value constructed in the previous step:

   ```
   /agent/ticket/1?TriggerAction=Ticket%3A%3ATicketFreeText
   ```

   Note: In case of custom URL query parameter values, we have to take care to encode all special characters that are normally used in URLs (e.g. double quotes). This process is called “URL encoding” , and can be done by any of freely available on-line tools, like the Online Text Tools.

When visiting the URL as constructed above, the action will be automatically triggered:

Note: Please keep in mind that any restrictions applied to the regular screen actions will still be honored. For example, you will not be able to trigger an action which is currently not accessible, e.g. because you don’ t have permissions to it, or it was filtered out by the ACL rules in effect.

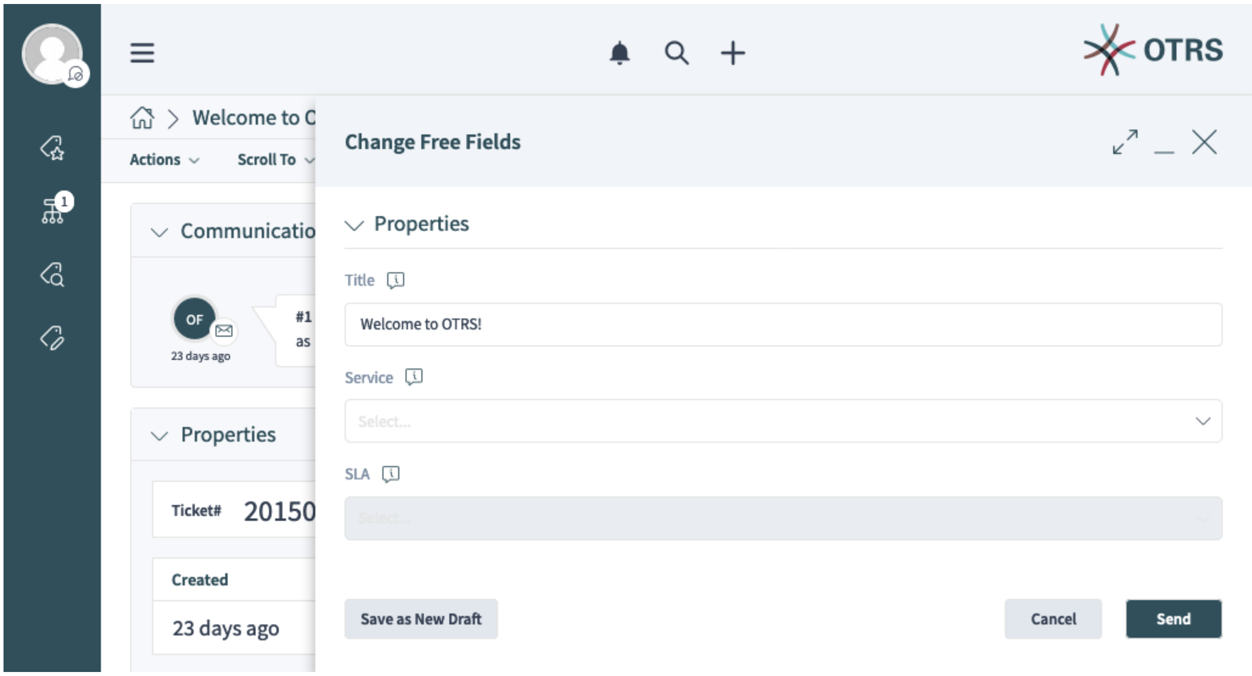
11.3.2 Pre-populating Form Fields

OTRS form fields can be pre-populated with values supplied via URL query parameters. The approach outlined below works for any form in the agent and external interface.

For example, let’ s see how we can pre-populate the *Title* field of the *Change Free Fields* action form, which we triggered in the previous example.

Since this action is not initially shown on the screen, we will keep the *trigger action* parameter in place, and just add on to it:

```
/agent/ticket/1?TriggerAction=Ticket%3A%3ATicketFreeText
```



Change Free Fields

Properties

Title: Welcome to OTRS!

Service: Select...

SLA: Select...

Save as New Draft

Cancel

Send

Fig. 1: Automatically Triggered Change Free Fields Action

1. Identify the name of the form field in question.

   In our case, we are dealing with the following form:

   ```
   Forms###AgentFrontend::Ticket::Action::FreeText
   ```

   According to *the form fields reference*, field is aptly named:

   ```
   Title
   ```

2. Append a query parameter to the URL of the detail view with the name of the field, and set it to the desired value:

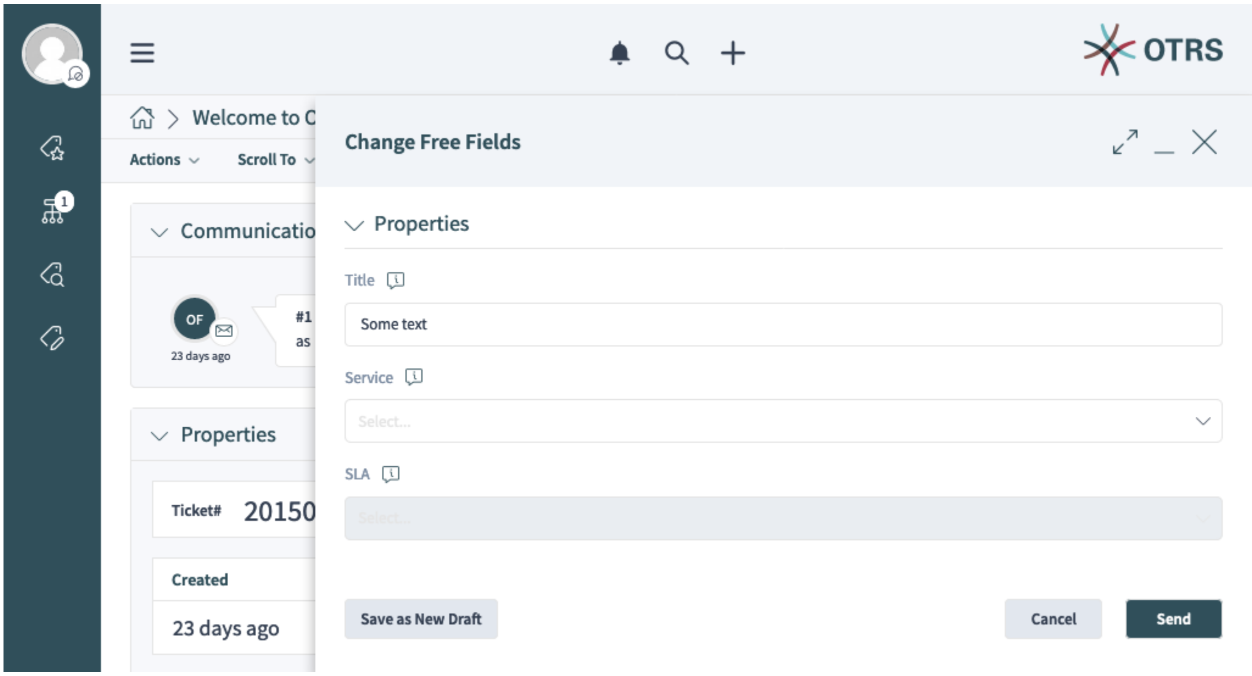
   ```
   /agent/ticket/1?TriggerAction=Ticket%3A%3ATicketFreeText&Title=Some+text
   ```

   Note: Additional URL query parameters can be chained with an ampersand character (&).

   Note: In case of custom URL query parameter values, we have to take care to encode all special characters that are normally used in URLs (e.g. double quotes). This process is called "URL encoding" , and can be done by any of freely available on-line tools, like the Online Text Tools.

When visiting the URL as constructed above, the action will be automatically triggered and the field pre-populated:

The approach can be repeated for all other fields in the form, for example, if we would also like to set the *Service* field (field name `ServiceID`) to a hypothetical value **Service 2** with an ID of **2**, we can add it to the existing URL like this:



Change Free Fields

Properties

Title: Some text

Service: Select...

SLA: Select...

Fig. 2: Pre-populated Title Field of the Change Free Fields Action

```
/agent/ticket/1?TriggerAction=Ticket%3A%3ATicketFreeText&Title=Some+text&
↪ServiceID=2
```

When visiting the URL as constructed above, the action will be automatically triggered and both fields pre-populated:

Note: In case of most drop-down fields that contain business objects (e.g. services, queues, etc.), you must always provide the ID of the object, not its label. You can recognize these form fields by the `ID` suffix in their name. You should always do a lookup of the object ID value, and set the field to it.

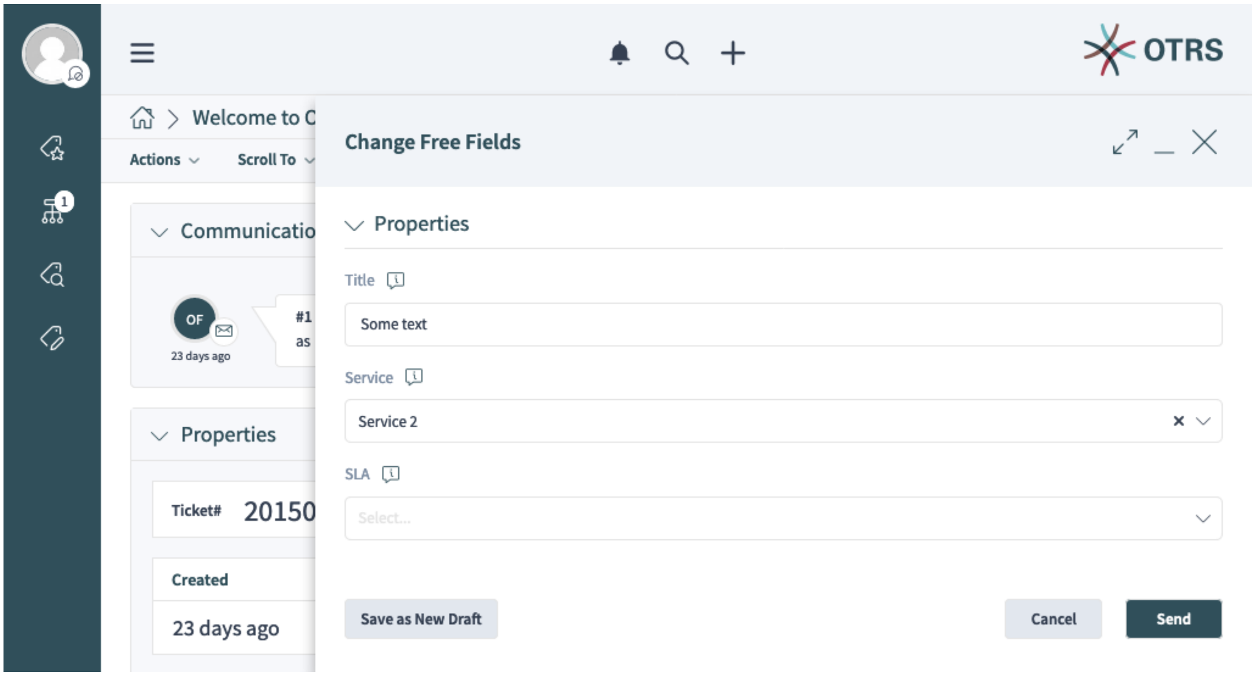
Note: Please keep in mind that any restrictions applied to the regular action form fields will still be honored. For example, you will not be able to set a value to a disabled field, or select an option which was filtered out by the ACL rules in effect.

11.3.3 Submitting Actions

Actions in the agent interface can be automatically submitted by via a special URL parameter called `SubmitForm`. The approach outlined below works for any action which is registered for the current screen.

For example, let's see how we can submit the *Change Free Fields* action form, which we already triggered and pre-populated in previous examples.

Since this action is not initially shown on the screen, we will keep the *trigger action* parameter in place, as well as some *form field parameters*, and just add on to it:



Title: Some text

Service: Service 2

SLA: Select...

Fig. 3: Pre-populated Title and Service Fields of the Change Free Fields Action

```
/agent/ticket/1?TriggerAction=Ticket%3A%3ATicketFreeText&Title=Some+text&
↪ServiceID=2
```

Append a query parameter to the URL of the detail view named `SubmitForm` and set **1** as its value:

```
/agent/ticket/1?TriggerAction=Ticket%3A%3ATicketFreeText&Title=Some+text&
↪ServiceID=2&SubmitForm=1
```

When visiting the URL as constructed above, the action will be automatically triggered, both fields pre-populated and the action immediately submitted:

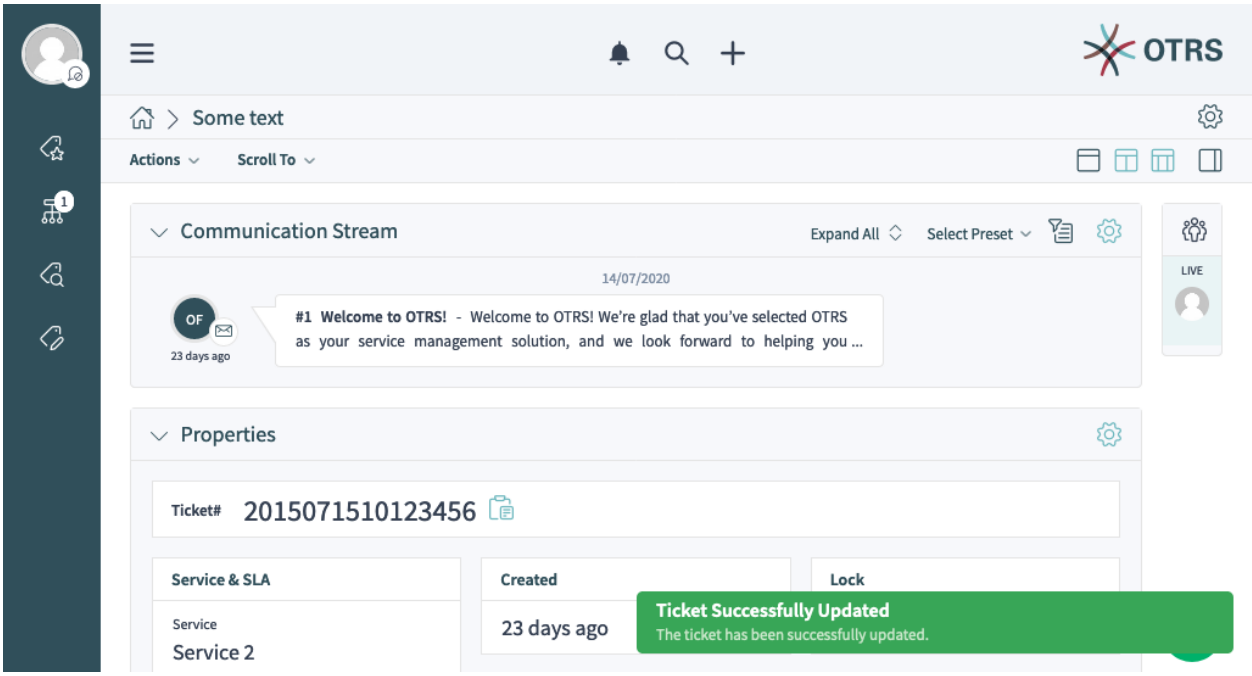
Note: Please keep in mind that any restrictions applied to the regular action forms will still be honored. For example, you will not be able to submit an invalid form by omitting a mandatory field, or setting a field to an invalid value.

11.3.4 Custom Business Object Lists

All static business object lists in the agent interface support custom configuration via a special URL parameter called `Config`. The approach outlined below works for similar screens which are listed in *the following section*, and for any keys documented in *the YAML reference*.

For example, let's see how we can set an active filter preset for a business object list by carefully constructing a direct URL for the view.

1. Identify the *slug name* of the view in question. We will be using `AgentFrontend::TicketList::Static` which has the slug:



14/07/2020

#1 Welcome to OTRS! - Welcome to OTRS! We're glad that you've selected OTRS as your service management solution, and we look forward to helping you ...

Ticket# 2015071510123456

Service & SLA	Created	Lock
Service Service 2	23 days ago	

Ticket Successfully Updated
The ticket has been successfully updated.

Fig. 4: Automatically Submitted Change Free Fields Action

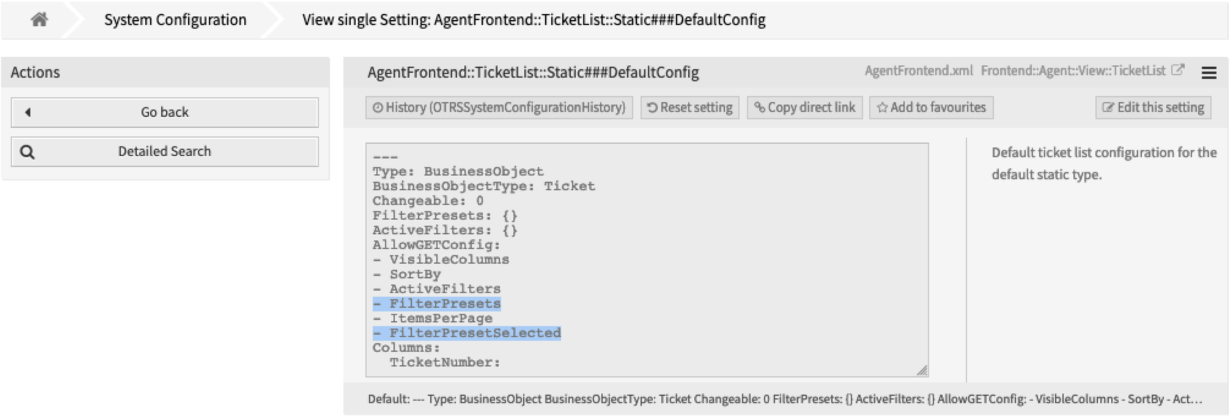
```
/agent/tickets/static
```

2. Go to *System Configuration* in the administrator interface and search for the screen configuration setting in question, in our case this is:

```
AgentFrontend::TicketList::Static###DefaultConfig
```

2. Make sure that the key `AllowGETConfig` in the screen configuration contains the properties `FilterPresets` and `FilterPresetSelected`, which we will pass via the query parameter.

   To be able change these keys, they must present in the list, otherwise they will be ignored. For more information, please see the *key reference*.



View single Setting: AgentFrontend::TicketList::Static###DefaultConfig

AgentFrontend::TicketList::Static###DefaultConfig

```
---
Type: BusinessObject
BusinessObjectType: Ticket
Changeable: 0
FilterPresets: {}
ActiveFilters: {}
AllowGETConfig:
- VisibleColumns
- SortBy
- ActiveFilters
- FilterPresets
- ItemsPerPage
- FilterPresetSelected
Columns:
  TicketNumber:
```

Default ticket list configuration for the default static type.

Default: --- Type: BusinessObject BusinessObjectType: Ticket Changeable: 0 FilterPresets: {} ActiveFilters: {} AllowGETConfig: - VisibleColumns - SortBy - Act...

Fig. 5: Screen Configuration with Defined Allowed Keys

3. Construct a valid YAML configuration that defines a user-defined filter preset and sets it as pre-selected.

```
---
FilterPresets:
  "Total Tickets":
    CustomerID:
      Value: my-customer@otrs.com
FilterPresetSelected: "Total Tickets"
```

4. Convert the YAML structure from previous step into JSON syntax, since YAML is unsuitable for passing via parameters. If we do this, we get the following analogous structure.

```
{
   "FilterPresets": {
      "TotalTickets": {
         "CustomerID": {
            "Value": "my-customer@otrs.com"
         }
      }
   },
   "FilterPresetSelected": "Total Tickets"
}
```

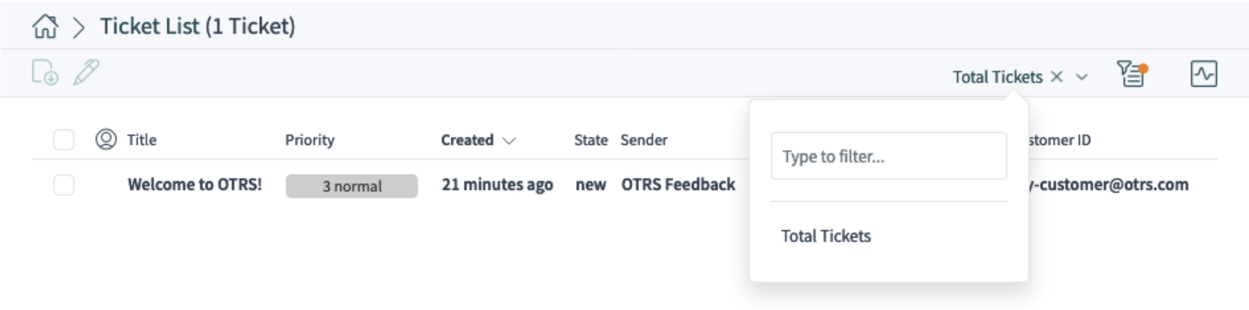
4. Remove all the extra white space to compress this configuration into a single line.

```
{"FilterPresets":{"Total Tickets":{"CustomerID":{"Value":"my-
↪customer@otrs.com"}}},"FilterPresetSelected":"Total Tickets"}
```

5. To pass the configuration to the screen located at `/agent/tickets/static`, we just need to define an URL query parameter named `Config` and set the JSON structure above as its value.

Note: In case of custom URL query parameter values, we have to take care to encode all special characters that are normally used in URLs (e.g. double quotes). This process is called "URL encoding" , and can be done by any of freely available on-line tools, like the Online Text Tools.

```
/agent/tickets/static?Config=%7B%22FilterPresets%22%3A%7B%22Total%20Tickets
↪%22%3A%7B%22CustomerID%22%3A%7B%22Value%22%3A%22my-customer%40otrs.com
↪%22%7D%7D%7D%2C%22FilterPresetSelected%22%3A%22Total%20Tickets%22%7D
```



Title	Priority	Created	State	Sender	stomer ID
Welcome to OTRS!	3 normal	21 minutes ago	new	OTRS Feedback	/-customer@otrs.com

Fig. 6: Example of the Filter Preset Defined via URL Query Parameter

Note: Did you know that you can use many freely available on-line tools to do the conversion steps outlined above? Some of them even support chaining, like the Online YAML Tools for the example above.

11.4 Inline Editing

OTRS provides an inline editing feature to easily modify any of the ticket attributes right in a list table or a property card. This allows for fast contextual edits, which can amplify an already instituted workflow for tickets and make it easier for agent users to do their daily work.

11.4.1 Inline Editing in Business Object Lists

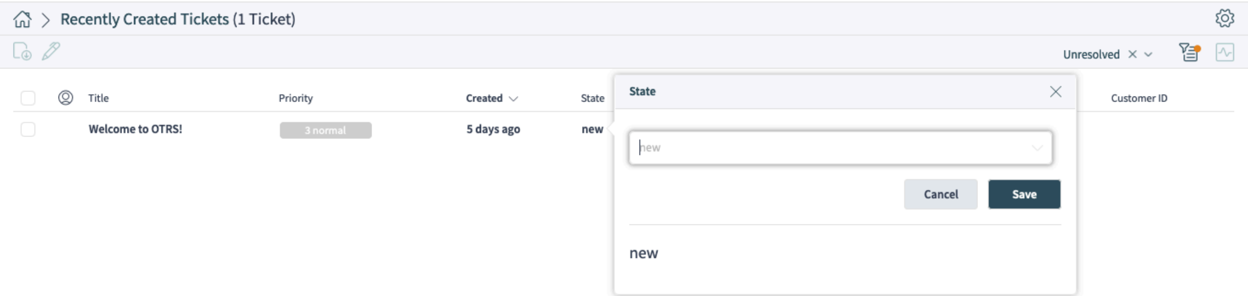
For an example, we will demonstrate how to turn on inline editing for the *State* ticket attribute in the *Recently Created Tickets* list screen.

1. Go to the *System Configuration* screen.
2. Search for the setting `AgentFrontend::TicketList::Created###DefaultConfig`.
3. Edit the setting and change the `IsInlineEditable` flag to `1` under the `State` column.

```
State:
  IsVisible: 2
  IsInlineEditable: 1
```

4. Search for the setting `AgentFrontend::Ticket::InlineEditing::Property###State` and verify the required permission level and lock state on the ticket, in order for the editing to be allowed for the user. By default, this attribute will require `state` permission type and ticket in a locked state.
5. Deploy the modified system configuration.

From now on, the *State* ticket attribute will be editable in a popover which is displayed by hovering the value in the same table column on the *Recently Created Tickets* list screen. By using the provided drop-down field, the ticket state can be changed after clicking on the *Save* button.



Title	Priority	Created	State	Customer ID
Welcome to OTRS!	3 normal	5 days ago	new	

State: new

Fig. 7: Ticket State Inline Editing in Business Object List

Warning: Please note that inline editing is only applicable for ticket lists in the organizer items or static screens, and not widgets. Ticket lists in widgets have very limited space, and the column popovers are disabled for them all together.

Note: For granular permission to the inline editing of ticket attributes, you can also employ *Access Control Lists (ACL)*. Each ticket attribute can be targeted by the `AgentFrontend::Ticket::InlineEditing::Property::*` endpoint namespace.

For our example from above, this can be done by the targeting the `AgentFrontend::Ticket::InlineEditing::Property::State` context.

11.4.2 Inline Editing in Property Cards

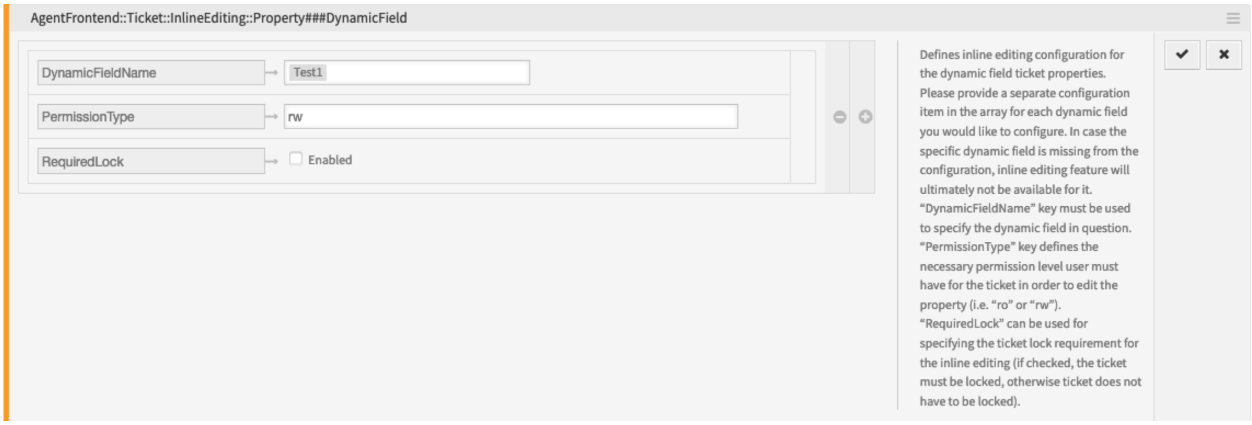
For an example, we will demonstrate how to turn on inline editing for a ticket dynamic field property card in the *Properties* widget on the ticket detail view.

The following example uses a dynamic field named `Test1`. Please make sure that you replace it with the actual name of your dynamic field.

1. Go to the *System Configuration* screen.
2. Search for the setting `AgentFrontend::TicketDetailView::WidgetType###Properties`.
3. Edit the setting and add a property definition under the `Properties` key for the dynamic field in question:

   ```
   Properties:
   - Name: DynamicField_Test1
     IsVisible: 2
     IsInlineEditable: 1
   ```

   Note the `IsInlineEditable` flag which is set to `1` in order to activate inline editing feature for this property card.
4. Search for the setting `AgentFrontend::Ticket::InlineEditing::Property###DynamicField`.
5. Edit the setting and add a configuration for the dynamic field by clicking on the plus button. Choose the name of the dynamic field under the `DynamicFieldName` key. Verify the required permission level and lock state on the ticket, in order for the editing to be allowed for the user. By default, this attribute will require `rw` permission type and ticket in an unlocked state.



AgentFrontend::Ticket::InlineEditing::Property###DynamicField

DynamicFieldName → Test1

PermissionType → rw

RequiredLock → □ Enabled

Defines inline editing configuration for the dynamic field ticket properties. Please provide a separate configuration item in the array for each dynamic field you would like to configure. In case the specific dynamic field is missing from the configuration, inline editing feature will ultimately not be available for it. “DynamicFieldName” key must be used to specify the dynamic field in question. “PermissionType” key defines the necessary permission level user must have for the ticket in order to edit the property (i.e. “ro” or “rw”). “RequiredLock” can be used for specifying the ticket lock requirement for the inline editing (if checked, the ticket must be locked, otherwise ticket does not have to be locked).

Fig. 8: Dynamic Field Inline Editing Configuration

5. Deploy the modified system configuration.

From now on, the configured dynamic field property card will be editable in the *Properties* widget of the ticket detail view. An editing icon in the upper right corner of the property card will switch the card in the editing mode. The dynamic field value can then be changed changed after clicking on the *Save* button.

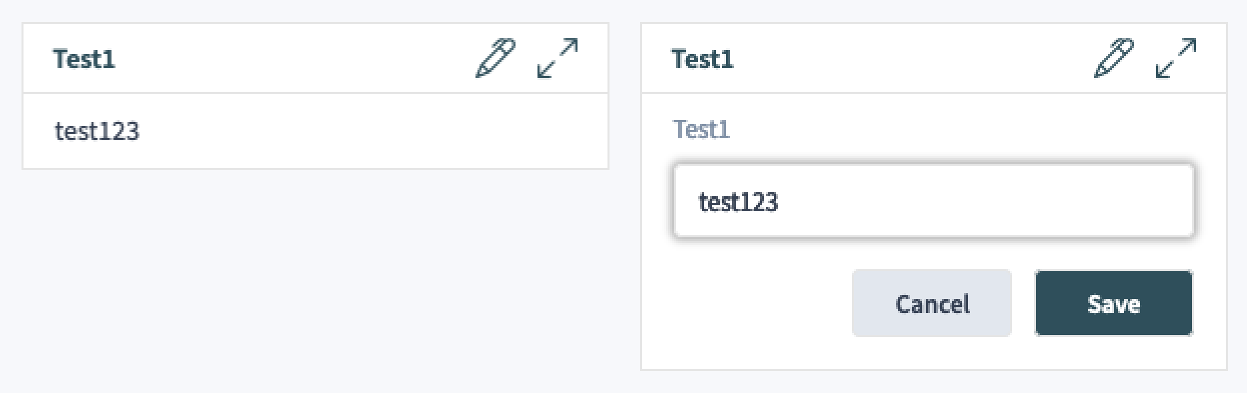


Fig. 9: Dynamic Field Inline Editing in Properties Widget

Warning: Please note that property cards are shown only for defined dynamic field values. If a ticket does not have a dynamic field value set, the property card will not be shown, and, in turn, it will not be possible to edit it inline. Similarly, in case you remove an existing dynamic field value, property card will disappear.

Note: For granular permission to the inline editing of ticket dynamic field values, you can also employ *Access Control Lists (ACL)*. Each dynamic field can be targeted by the `AgentFrontend::Ticket::InlineEditing::Property::DynamicField_*` endpoint namespace.

For our example from above, this can be done by the targeting the `AgentFrontend::Ticket::InlineEditing::Property::DynamicField_Test1` context.

11.5 External Link Previews

OTRS supports link previews via several features aimed at integrating externally provided content. By having contextual information in the right place, agent' s daily work can be made much more streamlined and straightforward.

11.5.1 Article Meta Filters

Article content can be automatically scanned and relevant information extracted with the help of meta filters.

For example, let' s configure extraction of CVE codes from article preview in the ticket detail view and display direct links to external pages with more information.

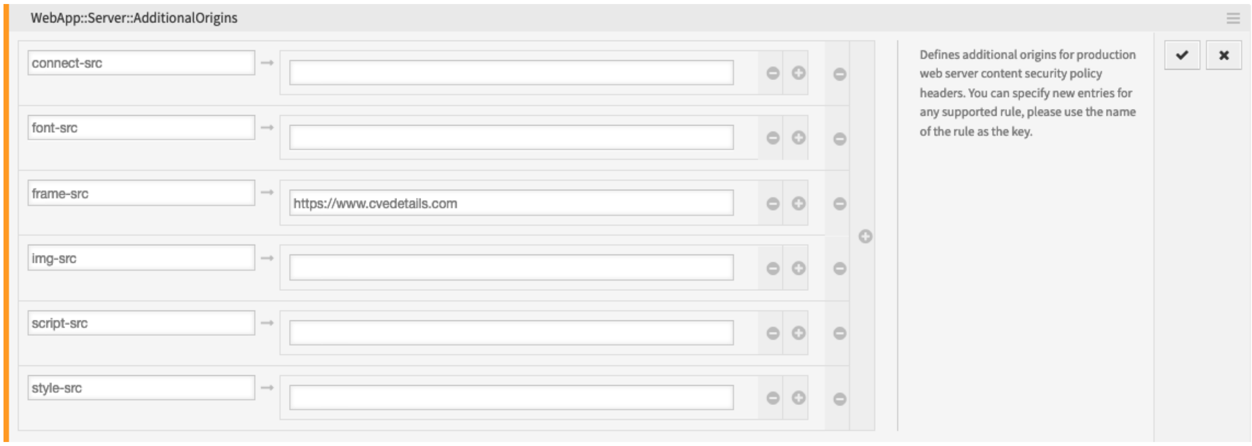
1. Go to the *System Configuration* in the administrator interface, and enable the `AgentFrontend::TicketDetailView::ArticleMeta` master switch for the article meta feature.

2. Search for the setting `AgentFrontend::TicketDetailView::ArticleMetaFilters###0002-Custom` and enable it.
3. Edit the setting, and set its value to the following YAML configuration:

```
---
- Label: CVE Details
  Name: cvedetails
  Target: _blank
  URL: https://www.cvedetails.com/cve/<MATCH1>-<MATCH2>-<MATCH3>
  URLPreview: https://www.cvedetails.com/cve/<MATCH1>-<MATCH2>-<MATCH3>
  RegExp:
  - (CVE|CAN)\-(\d{3,4})\-(\d{2,})
  Active: 1
```

4. To configure the Content Security Policy, search for the setting `WebApp::Server::AdditionalOrigins`.
5. Add the following origin under the `frame-src` key, in order to allow these web pages for embedding in the OTRS application:

```
https://www.cvedetails.com
```



WebApp::Server::AdditionalOrigins

connect-src	
font-src	
frame-src	https://www.cvedetails.com
img-src	
script-src	
style-src	

Defines additional origins for production web server content security policy headers. You can specify new entries for any supported rule, please use the name of the rule as the key.

Fig. 10: Additional Origins for Content Security Policy

6. Deploy the modified system configuration.

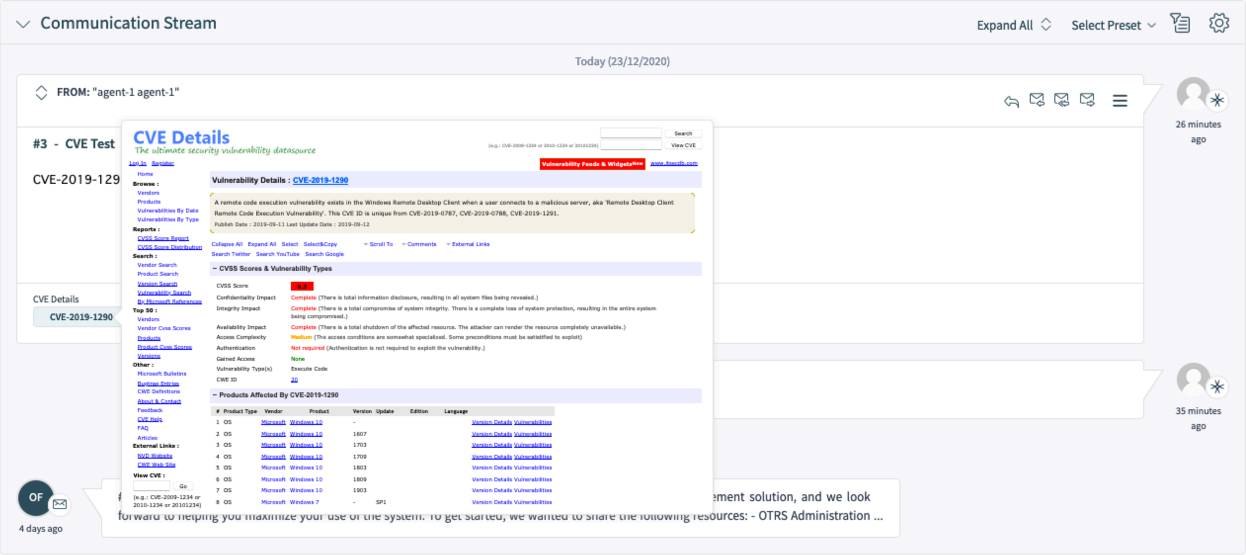
To test the feature, you will now need a ticket article that contains at least one valid CVE identifier in its text, for example `CVE-2019-1290`.

A button will be shown below the preview area in the *Communication Stream* or *Communication Compact* widget, which leads to the external page. Hovering over this button will show a preview popover with the scaled down content of the same page.

11.5.2 Dynamic Field Value

A dynamic field can be configured to show a link preview based on its value. The value can be used as whole or part of the link in question.

The following example use a dynamic field named `Field1`. Please make sure that you replace it with the actual name of your dynamic field.



Communication Stream

Expand All

Select Preset

Today (23/12/2020)

FROM: "agent-1 agent-1"

#3 - CVE Test

CVE-2019-129

CVE Details

CVE-2019-1290

26 minutes ago

35 minutes ago

OF

4 days ago

ement solution, and we look forward to helping you maximize your use of the system. To get started, we wanted to share the following resources: - OTRS Administration ...

CVE Details

The ultimate security vulnerability datasource

Log In Register

Search

(e.g.: CVE-2009-1234 or 2010-1234 or 20101234) View CVE

Vulnerability Feeds & WidgetsNew www.itsecdb.com

Home

Browse :

Vendors

Products

Vulnerabilities By Date

Vulnerabilities By Type

Reports :

CVSS Score Report

CVSS Score Distribution

Search :

Vendor Search

Product Search

Version Search

Vulnerability Search

By Microsoft References

Top 50 :

Vendors

Vendor Cvss Scores

Products

Product Cvss Scores

Versions

Other :

Microsoft Bulletins

Bugtraq Entries

CWE Definitions

About & Contact

Feedback

CVE Help

FAQ

Articles

External Links :

NVD Website

CWE Web Site

View CVE :

Go

(e.g.: CVE-2009-1234 or 2010-1234 or 20101234)

Vulnerability Details : CVE-2019-1290

A remote code execution vulnerability exists in the Windows Remote Desktop Client when a user connects to a malicious server, aka 'Remote Desktop Client Remote Code Execution Vulnerability'. This CVE ID is unique from CVE-2019-0787, CVE-2019-0788, CVE-2019-1291.

Publish Date : 2019-09-11 Last Update Date : 2019-09-12

Collapse All Expand All Select Select&Copy Scroll To Comments External Links

Search Twitter Search YouTube Search Google

– CVSS Scores & Vulnerability Types

CVSS Score	9.3
Confidentiality Impact	Complete (There is total information disclosure, resulting in all system files being revealed.)
Integrity Impact	Complete (There is a total compromise of system integrity. There is a complete loss of system protection, resulting in the entire system being compromised.)
Availability Impact	Complete (There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.)
Access Complexity	Medium (The access conditions are somewhat specialized. Some preconditions must be satistified to exploit)
Authentication	Not required (Authentication is not required to exploit the vulnerability.)
Gained Access	None
Vulnerability Type(s)	Execute Code
CWE ID	20

– Products Affected By CVE-2019-1290

#	Product Type	Vendor	Product	Version	Update	Edition	Language	
1	OS	Microsoft	Windows 10	-				Version Details Vulnerabilities
2	OS	Microsoft	Windows 10	1607				Version Details Vulnerabilities
3	OS	Microsoft	Windows 10	1703				Version Details Vulnerabilities
4	OS	Microsoft	Windows 10	1709				Version Details Vulnerabilities
5	OS	Microsoft	Windows 10	1803				Version Details Vulnerabilities
6	OS	Microsoft	Windows 10	1809				Version Details Vulnerabilities
7	OS	Microsoft	Windows 10	1903				Version Details Vulnerabilities
8	OS	Microsoft	Windows 7	-	SP1			Version Details Vulnerabilities

Fig. 11: Article Meta Preview

1. Configure the dynamic field to show a link based on its value via *Link for preview* setting. Please see *the setting reference* for more information.

   You can use the following value for testing:

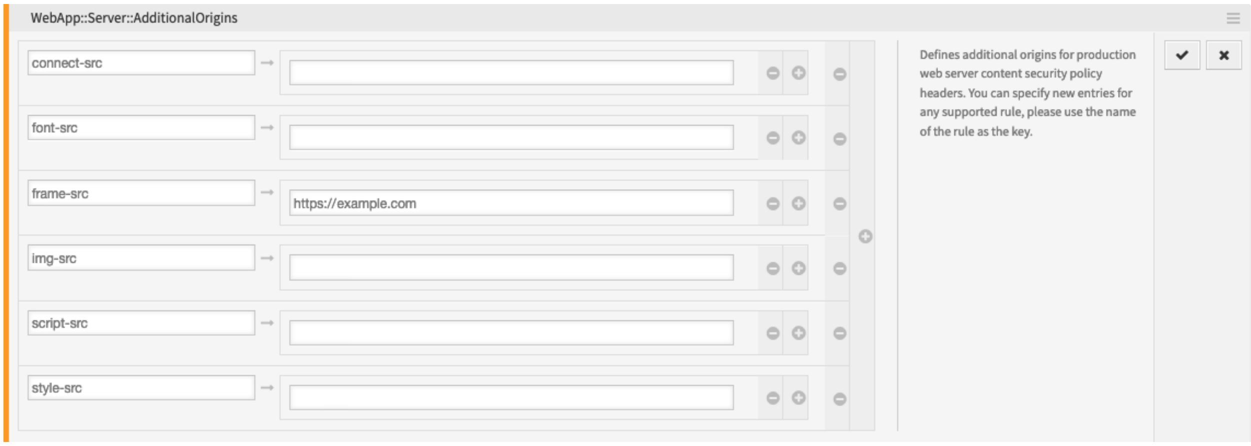
   ```
   https://example.com/handle?query=[% Data.Field1 | uri %]
   ```

2. Configure the display of the dynamic field value in ticket detail view, for example in the *Properties* widget. Please see *the relevant guide* for more information.

3. Search for the setting `WebApp::Server::AdditionalOrigins` in the *System Configuration*.

4. Add the following origin under the `frame-src` key, in order to allow these web pages for embedding in the OTRS application:

   ```
   https://example.com
   ```



WebApp::Server::AdditionalOrigins

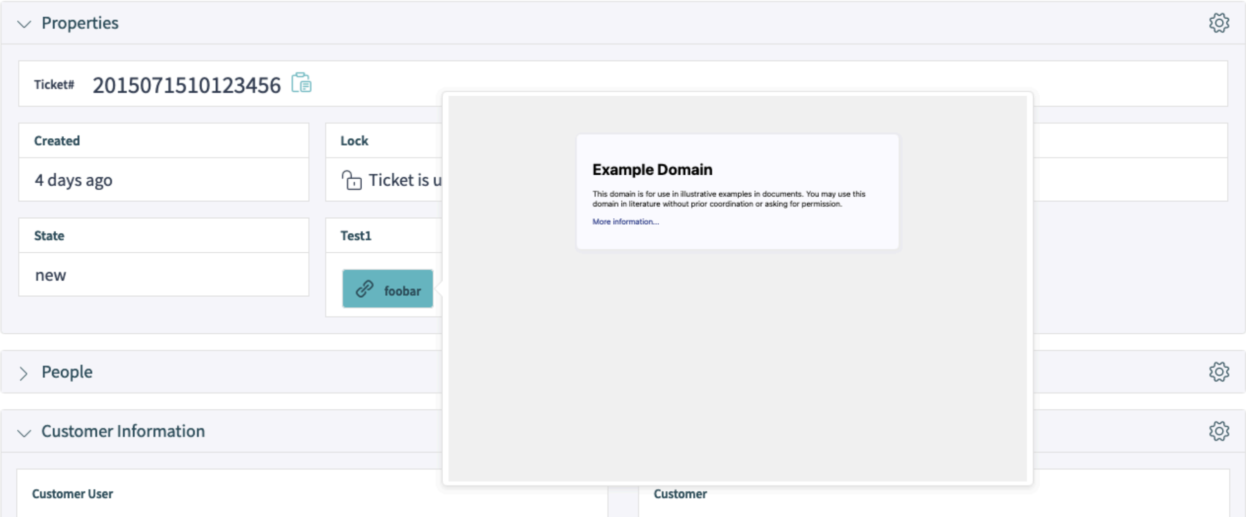
Key	Value
connect-src	
font-src	
frame-src	https://example.com
img-src	
script-src	
style-src	

Defines additional origins for production web server content security policy headers. You can specify new entries for any supported rule, please use the name of the rule as the key.

Fig. 12: Additional Origins for Content Security Policy

5. Deploy the modified system configuration.

To test the feature, you will now need a ticket with a value set for the dynamic field in question. To set this field via a ticket action, please see *the relevant guide* for more information.



Properties

Ticket# 2015071510123456

Created: 4 days ago

Lock: Ticket is u

State: new

Test1: foobar

Example Domain

This domain is for use in illustrative examples in documents. You may use this domain in literature without prior coordination or asking for permission.

More information...

People

Customer Information

Customer User

Customer

Fig. 13: Dynamic Field Link Preview

If defined, the ticket dynamic field value will be shown as a property card in the *Properties* widget, in form of a button which leads to the configured link. Hovering over this button will show a preview popover with the scaled down content of the same page.

11.5.3 Troubleshooting

In case you can not see the preview of the configured link, please note that target web servers might have a configuration in place that prevents them from being embedded in external pages. This is indicated by following or similar errors in the browser console:

```
Load denied by X-Frame-Options: “SAMEORIGIN” from “site”, site does not␣
↪permit cross-origin framing from “site”.
```

```
Firefox prevented this page from loading in this context because the page has␣
↪an X-Frame-Options policy that disallows it.
```

The blocking is normally done via the `X-Frame-Options` response header which can be set to the prevent the preview on the browser level. If you have the control of the target web server configuration, you might be able to relax this restriction by including the OTRS domain name in the list of allowed origins, or lift the restriction all together.

For more information, please take a look at X-Frame-Options page in the *Mozilla Developer Network*.

Alternatively, the newer `frame-ancestors` directive of the *Content Security Policy* header can be used to block embedding of the target server. This is indicated by following or similar error in the browser console:

```
Refused to display 'https://example.com/...' in a frame because an ancestor␣
↪violates the following Content Security Policy directive: "frame-ancestors
↪'self'".
```

For more information, please take a look at CSP: frame-ancestors page in the *Mozilla Developer Network*.

CHAPTER 12

Glossary

Activity Part of the process management.

Agent An agent is a user which services customer users. The person working for your organization.

Back end A configuration file, module, script or other functional item within OTRS which is not reachable via the browser.

Calendar Defines working hours, time-zone, and holidays to be applied when using *escalation times*.

Customer Physical organizations to which customer users are assigned. These provide the function to group users together which belong to a single organization.

Customer User Physical persons, and their contact data, which belong to a customer. These are the people raising tickets.

Dynamic Field A field used in tickets and articles to extend the data which can be saved in a ticket or article. They are saved from values in the database or accessed from outside resources.

Escalation Times The defined amount of working time before a ticket breaches an agreement.

Front end The graphical interface to OTRS as viewed in a browser.

Group This is a resource within OTRS. An *agent* or a *customer* can be assigned permissions to this resource as needed. Additionally, they can be used for access control, processes and web services.

Invoker An invoker is a special Perl module which allows OTRS to provide information to a remote system via REST or HTTP. Invokers must be developed in Perl by a back end developer.

Mapping A mapping allows OTRS to provide inbound and outbound translation of data allowing us to offer a specific data construction to rebuild incoming structures to meet our needs.

Operation Similar to a web hook, a provider operation offers a specific sub-set of OTRS Perl API functionality to external systems.

Process Part of the process management.

Process Path Part of the process management.

Resource A resource is any object which has effective permissions. This can be, but is not limited to, a ticket, dashboard widget, statistic, or module.

Screen A graphical interface to OTRS viewed in a browser (see also *Front end*).

Service A means of delivering added value to customers by making it easier for customers or helping them to achieve their desired results without them having to personally bear the responsibility for specific costs and risks.

Service Level Agreement A service level agreement (SLA) is a contract between a service provider (either internal or external) and the end user that defines the level of service expected from the service provider. SLAs are output-based in that their purpose is specifically to define what the customer will receive.

Service Request (see also *Ticket*).

Sequence Flow Part of the process management.

Sequence Flow Action Part of the process management.

SLA (see also *Service Level Agreement*).

Ticket A ticket is a collection of all communications with a customer during the course of a service request. A ticket contains article, which are the communication received from or sent to customers, agents, external systems, etc. Tickets belong to a customer user, are assigned to agents and reside in queues.

Transport A transport is the method chosen for communication. OTRS support REST and SOAP.

User Task Activity Dialog Part of the process management.

Working Time A definition of minutes passed during the defined working hours of any *calendar*.

Index