



OTRS Feature Add-ons Manual

Release 8.0

OTRS AG

Jan 20, 2021

Contents

1	Introduction	3
2	Advanced Escalations	5
2.1	Administrator Interface	6
2.2	Agent Interface	26
2.3	External Interface	31
3	Advanced Generic Agent	33
3.1	Administrator Interface	34
3.2	Agent Interface	34
3.3	External Interface	34
4	Advanced Ticket Article Forward	35
4.1	Administrator Interface	36
4.2	Agent Interface	36
4.3	External Interface	37
5	Agent Email Interface	39
5.1	Administrator Interface	40
5.2	Agent Interface	40
5.3	External Interface	41
6	Automated FAQ Ticket Creator	43
6.1	Administrator Interface	44
6.2	Agent Interface	49
6.3	External Interface	51
7	Automatic Start of Processes	53
7.1	Administrator Interface	54
7.2	Agent Interface	55
7.3	External Interface	56
8	Calendar Resource Planning	57
8.1	Administrator Interface	58
8.2	Agent Interface	66
8.3	External Interface	71
9	CI Assignment Attribute Dynamic Field Map	73

9.1	Administrator Interface	74
9.2	Agent Interface	76
9.3	External Interface	76
10	CI References	77
10.1	Administrator Interface	78
10.2	Agent Interface	81
10.3	External Interface	82
11	CIs in Customer Frontend	83
11.1	Administrator Interface	84
11.2	Agent Interface	86
11.3	External Interface	86
12	Configuration Management Connector	91
12.1	Administrator Interface	91
12.2	Agent Interface	97
12.3	External Interface	97
13	Custom Contact Fields	101
13.1	Administrator Interface	102
13.2	Agent Interface	105
13.3	External Interface	105
14	Customer Frontend Link Object	107
14.1	Administrator Interface	107
14.2	Agent Interface	108
14.3	External Interface	108
15	Customer-specific Services	109
15.1	Administrator Interface	110
15.2	Agent Interface	113
15.3	External Interface	113
16	Dashboard News Widget	115
16.1	Administrator Interface	115
16.2	Agent Interface	118
16.3	External Interface	119
17	Data Privacy Protection	121
17.1	Administrator Interface	122
17.2	Agent Interface	136
17.3	External Interface	136
18	Delete Attachments	137
18.1	Administrator Interface	137
18.2	Agent Interface	139
18.3	External Interface	139
19	Dynamic Field Attachment	141
19.1	Administrator Interface	141
19.2	Agent Interface	143
19.3	External Interface	145
20	Dynamic Field Calculation	147
20.1	Administrator Interface	147

20.2	Agent Interface	148
20.3	External Interface	149
21	Dynamic Field CI	151
21.1	Administrator Interface	152
21.2	Agent Interface	155
21.3	External Interface	155
22	Dynamic Field Database	157
22.1	Administrator Interface	158
22.2	Agent Interface	161
22.3	External Interface	162
23	Dynamic Field Web Service	163
23.1	Administrator Interface	164
23.2	Agent Interface	172
23.3	External Interface	172
24	Dynamic Sender Addresses	173
24.1	Administrator Interface	174
24.2	Agent Interface	177
24.3	External Interface	179
25	Escalation Suspend	181
25.1	Administrator Interface	182
25.2	Agent Interface	182
25.3	External Interface	183
26	Extended Ticket Stats	185
26.1	Administrator Interface	186
26.2	Agent Interface	186
26.3	External Interface	189
27	Hide/Show Dynamic Fields	191
27.1	Administrator Interface	192
27.2	Agent Interface	198
27.3	External Interface	198
28	Link Object Connector	199
28.1	Administrator Interface	199
28.2	Agent Interface	208
28.3	External Interface	208
29	Out of Office	209
29.1	Administrator Interface	210
29.2	Agent Interface	210
29.3	External Interface	210
30	Process Management Article Email	211
30.1	Administrator Interface	212
30.2	Agent Interface	213
30.3	External Interface	213
31	Ready2Adopt ITSM Processes	215
31.1	Administrator Interface	216
31.2	Agent Interface	217

31.3	External Interface	217
32	Ready2Adopt Processes	219
32.1	Administrator Interface	220
32.2	Agent Interface	220
32.3	External Interface	220
33	Ready2Adopt Web Services	221
33.1	Administrator Interface	222
33.2	Agent Interface	222
33.3	External Interface	222
34	Restore Pending Information	223
34.1	Administrator Interface	224
34.2	Agent Interface	224
34.3	External Interface	224
35	Restrict Customer Data View	225
35.1	Administrator Interface	226
35.2	Agent Interface	228
35.3	External Interface	228
36	Service-based Queue Routing	229
36.1	Administrator Interface	230
36.2	Agent Interface	232
36.3	External Interface	232
37	Service Categories	233
37.1	Administrator Interface	234
37.2	Agent Interface	236
37.3	External Interface	236
38	State Pre-selection Response Templates	237
38.1	Administrator Interface	238
38.2	Agent Interface	239
38.3	External Interface	239
39	System Configuration History	241
39.1	Administrator Interface	241
39.2	Agent Interface	243
39.3	External Interface	245
40	System Monitoring	247
40.1	Administrator Interface	247
40.2	Agent Interface	249
40.3	External Interface	249
41	Ticket Forms	251
41.1	Administrator Interface	252
41.2	Agent Interface	265
41.3	External Interface	266
42	Ticket Invoker	267
42.1	Administrator Interface	267
42.2	Agent Interface	272
42.3	External Interface	272

43 Ticket Queue Selection	273
43.1 Administrator Interface	274
43.2 Agent Interface	274
43.3 External Interface	274
44 Ticket Watchlist	275
44.1 Administrator Interface	276
44.2 Agent Interface	277
44.3 External Interface	281

This work is copyrighted by OTRS AG (<https://otrs.com>), Zimmersmühlenweg 11, 61440 Oberursel, Germany.

CHAPTER 1

Introduction

This manual lists all feature add-ons.

Note: The feature add-ons are installed into **OTRS** by the *Customer Solution Team*. In case of *On-Premise* systems, the customer can install the feature add-ons from the package manager, when the *Customer Solution Team* added the selected feature add-ons to the repository. To install a feature add-on, please contact the *Customer Solution Team* via support@otrs.com or in the [OTRS Portal](#).

Advanced Escalations

This feature add-on makes your escalation management more flexible and adjusts it according to your customers or to different service level agreements. Escalation types defined in **OTRS**, such as *First Response Time*, *Update Time*, and *Solution Time*, can be enhanced by creating new types and defining your own names and properties.

The *Ticket Escalation Types* option in the administrator interface enables you to define when escalations should:

- start,
- stop,
- be suspended,
- be resumed
- and be restarted.

Ticket attributes, like its status or certain events like creating or answering a ticket, can be used as a trigger. For example, an escalation can start when a ticket is created, but the escalation can stop when the ticket is answered by the agent. If the status of a ticket is changed to *Pending Reminder*, the escalation is suspended, but if the status is changed back to *Open*, the escalation resumes. An up-to-date display of the escalation time left makes accurate service time management possible.

In the ticket detail view, a new widget *Advanced Escalations* appears. This shows through the use of different color bars and numerical values whether or not:

- the escalation time is still within the original time frame (green),
- the escalation time will run out soon (orange),
- the escalation is paused (grey),
- the escalation time has been reached, i.e. the ticket has escalated (red) or
- escalation has been suspended or the ticket has been worked on during escalation (the widget is no longer visible).

The *Ticket Escalation Type Bundle* option enables you to assign newly created escalation types to different customers or service level agreements.

The following scenarios can now be handled more flexibly with *Advanced Escalations*:

- A customer requests a rework of a solution – the escalation must be adjusted.
- To present a solution, more information is required from the customer – the escalation must be suspended.
- A service technician can't get into the building or has no free access to the machine that needs to be fixed – the escalation must be suspended.
- And many more

Benefits

- Even more flexible escalation management – individually adoptable to customer or SLA.
- More precise service time management by detailed indication of remaining time.

Target Groups

- Customer service organizations having many partners or suppliers
- External IT service providers
- Call centers
- Sales departments and sales companies
- Advertising or communications agencies

Available in Service Package

- PLATINUM

Package Name in OTRS Package Manager

- OTRSAdvancedEscalations

2.1 Administrator Interface

With this package you are able to define your own escalations. You can define fully customized escalation types which contain information about under what circumstances a ticket escalation will start, restart, be suspended, be resumed or stopped.

After installation of the package two new modules will be available in the *Ticket Settings* group of the administrator interface.

2.1.1 Ticket Settings

After installation of the package two new modules will be available in the *Ticket Settings* group of the administrator interface and a new field is added to the *Service Level Agreements* module.

Note: The configuration of the standard escalations in OTRS are not related to advanced escalations. You should decide if you want to use the standard escalation types of OTRS or individual escalation types of advanced escalations. If you configure both then you will have parallel escalations.

Escalation Type Bundles

The *Escalation Types* can be grouped to so-called bundles. Bundles can be connected to SLAs, customers, priorities and services in order to define special escalations on a per-customer basis and depending on certain calendar settings.

After installation of the package a module *Escalation Type Bundles* will be available in the *Ticket Settings* group of the administrator interface.

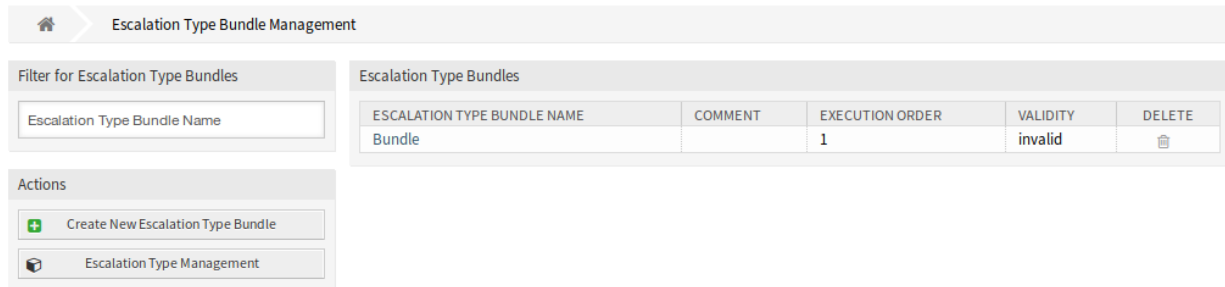


Fig. 1: Escalation Type Bundle Management Screen

The advanced escalations package works with the relationships chain *Escalation type* → *Escalation type bundle* → *SLA*.

One or more SLA escalation type should be created and assigned to an escalation type bundle, and one or more escalation bundles should be related to one or more SLAs. The relationship between *Customer* → *Service* → *SLA* is the normal one on the system.

Manage Escalation Type Bundles

To create a new escalation type bundle:

1. Click on the *Create New Escalation Type Bundle* button in the left sidebar.
2. Fill in the required fields.
3. Click on the *Create* button.
4. You will be redirected to *Edit Escalation Type Bundle* screen to edit the escalation type bundle details.

To edit an escalation type bundle:

1. Click on an escalation type bundle in the list of escalation types bundles or you are already redirected here from *Create New Escalation Type Bundle* screen.
2. Modify the fields and the escalation type bundle details.
3. Click on the *Save* or *Save and finish* button.

To delete an escalation type bundle:

1. Click on the trash icon in the *Delete* column.
2. Click on the *OK* button in the confirmation dialog.

Create New Escalation Type Bundle

★ Name:

Comment:

Description:

Customers:

Priorities:

Services:

★ Execution Order:

★ Validity:

or Cancel

Fig. 2: Create New Escalation Type Bundle Screen

Escalation Type Bundle 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.

Customers Select a customer from the drop-down list. If one or more customers are selected, the bundle will only take effect on tickets which are assigned to one of the selected customers.

Priorities Select a priority from the drop-down list. If one or more priorities are selected, the bundle will only take effect on tickets which matches one of the selected priorities.

Services Select a service from the drop-down list. If one or more services are selected, the bundle will only take effect on tickets which matches one of the selected services.

Execution order * Select, in which order should the escalation bundles be evaluated.

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.

Add escalation type Bundles can contain several escalation types. On a bundle, all information regarding the time for the escalation is being stored.

Escalate after Define the escalation time span and unit.

Edit Escalation Type Bundle Bundle

★ Name:

Comment:

Description:

Customers:

Priorities:

Services:

★ Execution Order:

★ Validity:

Add escalation type:

Escalation type "First Response" ✕

Escalate after: Round up times

Use calendar: Notify on: %

or or [Cancel](#)

Fig. 3: Edit Escalation Type Bundle Screen

Escalation Type Bundles

ESCALATION TYPE BUNDLE NAME	COMMENT	EXECUTION ORDER	VALIDITY	DELETE
Bundle		1	invalid	

Fig. 4: Delete Escalation Type Bundle Screen

Round up times Check this box to round up the times, so the start of the escalation will take place at the next full time unit.

For example if *1 hour* is set as escalation time and a ticket is being created at 10:23, the escalation will start at 11:00, rounding up to the next full hour.

Use calendar Select a calendar to used by the escalation. If no calendar is being used, a 24/7/365 calendar without weekends or other free days will be assumed.

Notify on Define the notification time.

Escalation Types

Escalation types contain generic information about the conditions that need to be matched in order to start, restart, suspend, resume or stop an escalation on a ticket. The type itself does not contain any time settings (e.g. how many hours need to pass until the escalation starts).

Use this screen to add escalation types to the system. After installation of the package some escalation types are already added. The escalation type management screen is available in the *Escalation Types* module of the *Ticket Settings* group.

ESCALATION TYPE NAME	COMMENT	VALIDITY	EXPORT	DELETE
First Response		valid		
Solution Time		valid		
Update Time		valid		

Fig. 5: Escalation Type Management Screen

Manage Escalation Types

To create a new escalation type:

1. Click on the *Create New Escalation Type* button in the left sidebar.
2. Fill in the required fields.
3. Click on the *Create* button.
4. You will be redirected to *Edit Escalation Type* screen to edit the escalation type details.

To edit an escalation type:

Create New Escalation Type

★ Name:

Comments:

Description:

Allow resume from stop:

★ Validity:

or

Fig. 6: Create New Escalation Type Screen

1. Click on an escalation type in the list of escalation types or you are already redirected here from *Create New Escalation Type* screen.
2. Modify the fields and the escalation type details.
3. Click on the *Save* or *Save and finish* button.

To delete an escalation type:

1. Click on the trash icon in the *Delete* column.
2. Click on the *OK* button in the confirmation dialog.

To export an escalation type:

1. Click on the export icon in the *Export* column.
2. Choose a location in your computer to save the `Export_EscalationTypeID_X.yml` file.

To import an escalation type:

1. Click on the *Browse...* button in the left sidebar.
2. Select a previously exported `.yml` file.
3. Click on the *Import escalation type configuration* button.

Escalation 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.

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.

Edit Escalation Type First Response

★ Name:

Comments:

Description:

Allow resume from stop:

★ Validity:

Start Restart Suspend Resume Stop

Hint: Within this tab you can set up the conditions that need to be matched to Start the escalation. Please use the checkbox in the first column of each block to enable the condition and set it up afterwards. Please keep in mind that the different blocks and the conditions within the blocks are logically connected by 'and', while the sets of ticket attributes are logically connected by 'or'.

General

Use this dynamic field as base time for this event:

Ticket attributes [+ Add attribute set](#)

ACTIVE	CONDITION
<input checked="" type="checkbox"/>	A message from an/a <input type="text" value="Agent"/> is not present <input type="text"/>
and <input checked="" type="checkbox"/>	The ticket attribute <input type="text" value="StateType"/> did never equal <input type="text"/> one of these values: <input type="text" value="closed"/> <input type="text" value="removed"/> <input type="text" value="merged"/> <input type="text" value="pending auto"/> <input type="text" value="pending reminder"/> +
and <input checked="" type="checkbox"/>	The ticket attribute <input type="text" value="StateType"/> equals now <input type="text"/> one of these values: <input type="text" value="new"/> <input type="text" value="open"/> +

or or

Fig. 7: Edit Escalation Type Screen







Escalation Types				
ESCALATION TYPE NAME	COMMENT	VALIDITY	EXPORT	DELETE
First Response		valid		
Solution Time		valid		
Update Time		valid		

Fig. 8: Delete Escalation Type Screen

Allow resume from stop If this is checked, it allows this escalation to be resumed although it had already been stopped at some point in the past (of course only if the related resume conditions match).

Note: This setting will only apply on escalations which have not yet reached the escalation time, which means that you should not expect escalations which have already been stopped to resume after enabling the checkbox.

If the resume conditions match on an escalation type which has *Allow resume from stop* enabled, the escalation will be resumed with the time that was still left on this escalation when it was stopped (which means *was fulfilled*).

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.

Condition types

This widget has five tabs for *Start*, *Restart*, *Suspend*, *Resume* and *Stop* events. The following options are available for the tabs.

General

This setting is only available for the *Start* and *Restart* event of the escalation. This setting controls if the ticket creation time or the time of the current action should be taken as base time for the start or the restart of the escalation.

Example: If you set up *current time* as base time for the restart event, the full escalation time will be available (e.g. 1 hour) if the restart condition matches. If you set up *ticket creation time*, the escalation will restart based on the creation time of the ticket. This could be useful if you close a ticket as successful (which stops your escalation) but it turns out that the problem has not been solved.

You are also able to set a date or a date-time dynamic field as base time. If you create a ticket with this dynamic field filled and the escalation start then the value of the dynamic field will be used as base time. If the configured dynamic field is not filled then the fallback for the base time is used (current time or ticket creation time or no time value - escalation will only start if the dynamic field is filled).

Ticket Attributes

Several attribute sets can be set up. Each of these sets can contain several attribute conditions. The sets are logically connected by *OR*, which means that one of the sets has to be matched in order to trigger the event. The conditions within a set are logically connected by *AND*, which means that all of the conditions have to match in order to verify the whole set.

After adding a new set, you can add conditions to this set by using the *Add attribute set* button on the top right of the widget. After enabling the added condition in the widget, you can choose what ticket attribute you want to check.

Note: All configured conditions are used to evaluate values present on each action, which is defined as a snapshot of the ticket values taken from the ticket history. The time window for this snapshot is defined by the `TicketHistorySplitTimeThreshold` setting. All actions for a ticket are evaluated one by one each time the escalation is recalculated.

Ticket created by a customer This condition type means that the ticket has to be created by customer user (by sending an email, using the external interface or by an agent using the *New Phone Ticket* screen).

Ticket Attributes In this drop-down menu you can choose from a list of match types:

equals now (Is) – One of the configured ticket attributes should match the particular attribute value from the action which is currently being evaluated. The condition evaluates as true if there is at least one match.

doesn't equal now (Is not) – None of the configured ticket attributes should match the particular attribute value from the action which is currently being evaluated. The condition evaluates as true if there is no match.

equaled (Was) – One of the configured ticket attributes should match the particular attribute value from the action which is currently being evaluated. All actions from the history of the particular ticket are being evaluated. The condition evaluates as true if there is at least one match in one of the historical ticket attribute values.

did never equal (Was never) – None of the configured ticket attributes should match the particular attribute value from the action which is currently being evaluated. All actions from the history of the particular ticket are being evaluated. The condition evaluates as true if there is no match in one of the historical ticket attribute values.

changed to (Changed to) – The configured ticket attribute changed to one of the given values. This implies that the value was different before (= on the previous action).

equals previous (Was previously) – At least one of the configured ticket attributes should match the particular attribute value on the previous action, which means going back exactly one step in the ticket history. The condition evaluates as true if there is a match in one of the historical ticket attribute values of the previous action.

doesn't equal previous (Was not previously) – None of the configured ticket attributes should match the particular attribute value on the previous action, which means going back exactly one step in the ticket history. The condition evaluates as true if there is no match in the historical ticket attribute values of the previous action.

Message presence This condition type defines if a customer user or agent message must be/must not be present.

Message delivery If enabled, this will define that the customer user or agent has to send a message to match the condition.

To remove a condition, uncheck the checkbox in the *Active* column, and click on the *Save* button.

Service Level Agreements

It is possible to assign escalation type bundles to an SLA. Therefore, a multiselect field has been added to the *Add Service Level Agreement* screen.

Add SLA

★ SLA:

Service:

Calendar:

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

Comment:

Escalation Bundles :
Define the escalation bundles.

Dialog message:

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

or

Fig. 9: Add Service Level Agreement Screen

New Field

Escalation Bundles Here you can choose from a list of all available bundles which should be assigned to this SLA. Assigning a bundle to a SLA will cause the escalation types, which are assigned to this bundle, to be considered for a ticket which has this SLA set.

Note: The calendar setting in the SLA has no effect for the advanced escalations feature.

2.1.2 Communication & Notifications

After installation of the package there will be some new events and smart tags available in ticket notifications.

Ticket Notifications

This package adds some new features to the ticket notification methods.

Notification Smart Tags

The notification smart tags only work in notifications related to the escalation. It does not work for `TicketCreate` but it will work for *Escalation: First Response* (`NotifyBefore`). The following smart tags with information about each escalation are available:

<OTRS_TICKET_CustomEscalation_EscalationTime> If the escalation is running, then this tag will return the escalation date-time when the escalation type will escalate (e.g. *2019-01-01 10:00:00*).

<OTRS_TICKET_CustomEscalation_NotifyTime> If the escalation is running, then this tag will return the escalation notify date-time of the escalation type (e.g. *2019-01-01 10:00:00*).

<OTRS_TICKET_CustomEscalation_EscalationTimeIn> If the escalation is running, then this tag will return the difference the current time and escalation time in format *3h 30m*.

<OTRS_TICKET_CustomEscalation_TypeName> This tag will return the name of the escalation type (e.g. *solution time*).

Configure Event Based Notifications

Use this steps to configure event based notifications for users of escalated tickets:

1. Make sure OTRS daemon is running.

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

2. Go to the system configuration and enable `Daemon::SchedulerCronTaskManager::Task###TriggerEscalati` setting. It will allow the generic agent to run the module in charge to start the advanced escalations events.
3. Go to the administrator interface and open the *Ticket Notifications* module to configure the notifications.
4. Configure the notification and change text by using the new notification smart tags provided by this package and described above.

Escalation Events

The following escalation events are available:

Escalation: [EscalationName] (Start) (EscalationStart_[EscalationTypeID]) This event will be executed if the escalation has been started.

Escalation: [EscalationName] (NotifyBefore) (EscalationNotifyBefore_[EscalationTypeID]) This event will be executed if the time reached where the agent gets notified for the escalation.

This event will be only executed correctly if the escalation bundle configuration per escalation type is correctly configured, `TriggerEscalationStartEvent` is enabled and the daemon is running.

Escalation: [EscalationName] (Breached) (EscalationBreached_[EscalationTypeID]) This event will be executed if the escalation time has been reached.

This event will be only executed correctly if the escalation bundle configuration per escalation type is correctly configured, `TriggerEscalationStartEvent` is enabled and the daemon is running.

Escalation: [EscalationName] (Restart) (EscalationRestart_[EscalationTypeID]) This event will be executed if the escalation has been restarted.

Escalation: [EscalationName] (Suspend) (EscalationSuspend_[EscalationTypeID]) This event will be executed if the escalation has been suspended.

Escalation: [EscalationName] (ResumeSuspend) (EscalationResumeSuspend_[EscalationTypeID]) This event will be executed if the escalation has been resumed from a suspend state.

Escalation: [EscalationName] (ResumeStop) (EscalationResumeStop_[EscalationTypeID]) This event will be executed if the escalation has been resumed from a stop state.

Escalation: [EscalationName] (Stop) (EscalationStop_[EscalationTypeID]) This event will be executed if the escalation has been stopped.

2.1.3 Administration

Two new tables are added to the database after installation of the package. The new tables can be used for reporting via SQL box.

Additionally there are several new system configuration options available. You will find these new options in the group `OTRSAdvancedEscalations`.

SQL Box

Escalation history is available for SQL reporting. The following chapters explain the structure of the database tables.

`escalation_history` Table

All escalation events will create new entries in the `escalation_history` table, which is the basis to calculate statistics about the completed escalation cycles.

To activate the additional reporting of the escalation events, you need to enable the following system configuration option:

- `Ticket::EventModulePost###EscalationHistory` (group: `OTRSAdvancedEscalations`, navigation: `Core` → `Event` → `EscalationHistory`).

Make sure OTRS daemon is running.

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

This event module will track the following escalation events:

- EscalationStart
- EscalationStop
- EscalationSuspend
- EscalationRestart
- EscalationResumeSuspend
- EscalationResumeStop

The `escalation_history` table has the following columns:

id This column contains the ID of the escalation history (auto increment).

event_trigger This column contains the event of the escalation (e.g. `EscalationStart`).

object_id This column contains object ID of the escalation (e.g. the ticket ID).

object_type This column contains object type of the escalation (e.g. `Ticket`).

object_history_id This column contains the related ID of the `ticket_history` table to the escalation event.

escalation_type_id This column contains the escalation type ID of the escalation.

escalation_reached This column contains whether the escalation time is already reached (possible values: 0/1).

escalation_datetime This column contains the timestamp of the escalation date.

escalation_time This column contains the rest of time (seconds) until the ticket will escalate.

escalation_wt This column contains the rest of time (seconds) until the ticket will escalate (calculated with working calendars).

notify_datetime This column contains the date-time timestamp of the notify start.

notify_time This column contains the seconds until the notify start.

escalation_remaining_time This column contains the rest of time until the ticket will escalate after a suspend of an escalation type.

Note: This column is only filled on suspend state.

escalation_remaining_wt This column contains the rest of time until the ticket will escalate after a suspend of an escalation type (calculated with working calendars).

Note: This column is only filled on suspend state.

notify_remaining_time This column contains the seconds until the notify start after a suspend of an escalation type.

Note: This column is only filled on suspend state.

notify_remaining_wt This column contains the seconds until the notify start after a suspend of an escalation type (calculated with working calendars).

Note: This column is only filled on suspend state.

running_total_time This column contains the total amount of seconds the timer was running based on the `Timer (Start | Restart | Suspend | Resume | Stop)` events.

running_total_wt This column contains the total amount of seconds the timer was running based on the `Timer (Start | Restart | Suspend | Resume | Stop)` events (calculated with working calendars).

running_total_virtual_time This column contains the total amount of seconds the timer was running based history entries.

running_total_virtual_wt This column contains the total amount of seconds the timer was running based history entries (calculated with working calendars).

suspend_total_time This column contains the total amount of suspended seconds of the escalation type based on the `Timer (Start | Restart | Suspend | Resume | Stop)` events.

suspend_total_wt This column contains the total amount of suspended seconds of the escalation type based on the `Timer (Start | Restart | Suspend | Resume | Stop)` events (calculated with working calendars).

running_last_time This column contains the seconds between a start or resuming event and a stop or suspending event (e.g. `EscalationStart` to `EscalationSuspend` or `EscalationResume` to `EscalationStop`).

running_last_wt This column contains the seconds between a start or resuming event and a stop or suspending event (e.g. `EscalationStart` to `EscalationSuspend` or `EscalationResume` to `EscalationStop`) (calculated with working calendars).

running_last_virtual_time This column contains the seconds between a start or resuming event and a stop or suspending event (e.g. `EscalationStart` to `EscalationSuspend` or `EscalationResume` to `EscalationStop`) based on the history entries of the ticket.

running_last_virtual_wt This column contains the seconds between a start or resuming event and a stop or suspending event (e.g. `EscalationStart` to `EscalationSuspend` or `EscalationResume` to `EscalationStop`) based on the history entries of the ticket (calculated with working calendars).

suspend_last_time This column contains the amount of seconds the ticket escalation was suspended last time based on the history entries of the ticket.

suspend_last_wt This column contains the amount of seconds the ticket escalation was suspended last time based on the history entries of the ticket (calculated with working calendars).

create_time This column contains create time of the escalation history entry.

create_by This column contains ID of the user who triggered the history data set.

change_time This column contains date and time when the escalation history data set was changed.

change_by This column contains ID of the user who triggered the data set change.

escalation_history_data Table

All escalation events will create new entries in the `escalation_history` table. For each escalation event it is possible to save ticket and dynamic field data in a separate data table. Make sure that

`TriggerEscalationStartEvents` is enabled. The attributes which will be saved can be configured in the following system configuration options:

- `EscalationHistoryData###Ticket` (group: `OTRSAdvancedEscalations`, navigation: `Core` → `EscalationHistoryData`).

Example configuration: `Queue` → `1`

- `EscalationHistoryData###DynamicField` (group: `OTRSAdvancedEscalations`, navigation: `Core` → `EscalationHistoryData`).

Example configuration: `DynamicField_Test` → `1`

To activate the additional reporting of the escalation events, you need to enable the following system configuration option:

- `Ticket::EventModulePost###EscalationHistory` (group: `OTRSAdvancedEscalations`, navigation: `Core` → `Event` → `EscalationHistory`).

Make sure OTRS daemon is running.

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

The data of the ticket and dynamic fields will be saved in a separate table `escalation_history_data` with the following columns:

id This column contains the ID of the escalation history (auto increment).

escalation_history_id This column contains ID of the related `escalation_history` entry.

field_key This column contains key of the related data (e.g. `DynamicField_Test` or `Queue`).

field_value This column contains value of the related data (e.g. a dynamic field value or the values of ticket attributes).

create_time This column contains create time of the escalation history data entry.

create_by This column contains ID of the user who triggered the escalation history data set.

change_time This column contains date and time when the escalation history data set was changed.

change_by This column contains ID of the user who triggered the data set change.

System Configuration

In the ticket lists and also in the ticket list widgets, information regarding the escalation data is displayed in one column for each escalation type. These general columns will be made available to all users in all ticket lists by default.

For the general escalation columns, following traffic light colors for the escalation states are used:

- Green: No escalation times are reached or exceeds the limit.
- Orange: The warning time was reached or exceeds the limit, but no escalation time is reached or exceed the limit yet.
- Red: Escalation time is reached or exceeds the limit.
- Grey: Escalation time is currently paused.

In addition to this, you have the possibility to specify which advanced escalation columns you want to use in the ticket lists.

These possible advanced escalation columns are not active by default. If you want to use them, you have to switch them on explicitly. Please follow instructions as outlined in the sections below.

Advanced escalation columns are shown like this: *Escalation type (Column)*, e.g. *First Response Time (Escalation reached, yes/no)*. They are translatable.

Display Advanced Escalation Columns

It is possible to display the escalation times in the ticket lists widget and in the ticket lists.

Dashboard Ticket List Widgets

In the following example, we will add an advanced escalation column named `EscalationDatetime` to the *Escalations* widget on the dashboard, for all available escalation types.

1. Go to *System Configuration* screen.
2. Search for the setting `AgentFrontend::Dashboard::Widget###EscalatedTickets`.
3. Under the `Config` key, add the following keys to the existing YAML configuration:

```

...
Config:
  ...
  Columns:
    ...
    EscalationType_EscalationDatetime:
      IsVisible: 2
  
```

4. Make sure to just **add** the new column name to the existing structure, taking care to follow the rules of the YAML syntax.

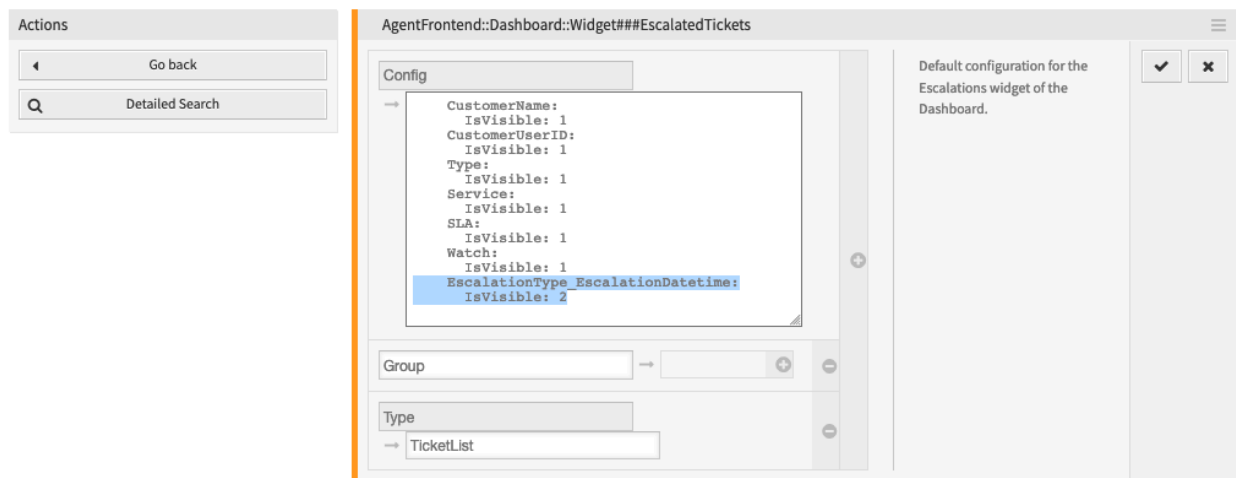


Fig. 10: Add Advanced Escalation Column to *Escalations* Widget Configuration

Note: To make the column available to users, but not visible by default, switch `IsVisible` key to 1.

- Click on the tick button on the right to save the setting.
- Deploy the modified system configuration.

The referenced advanced escalation column will now be visible by default in the *Escalations* widget, for all available escalation types.

Priority	Created	First Response (Escalation reach date and time)	Update Time (Escalation reach date and time)	Solution Time (Escalation reach date and time)
3 normal	5 days ago	5 days ago	2 hours ago	5 days ago

Fig. 11: Advanced Escalation Columns in the *Escalations* Widget on Dashboard

Ticket List Screens

In the following example, we will add an advanced escalation column named `EscalationReached` to the *Escalated Tickets* screen, for all available escalation types.

- Go to *System Configuration* screen.
- Search for the setting `AgentFrontend::TicketList::Escalations###DefaultConfig`.
- Under the `Config` key, add the following keys to the existing YAML configuration:

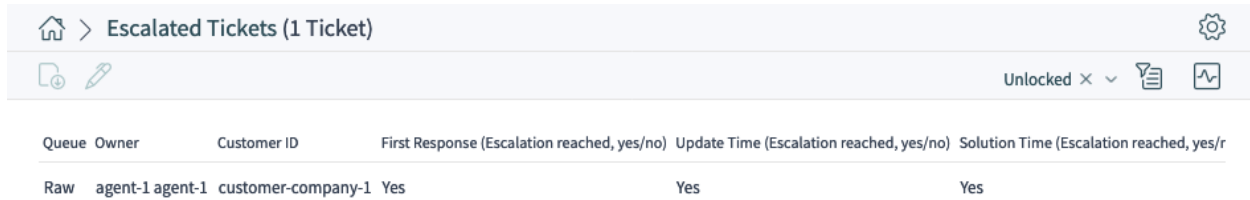
```
...
Columns:
  ...
  EscalationType_EscalationReached:
    IsVisible: 2
```

- Make sure to just **add** the new column name to the existing structure, taking care to follow the rules of the YAML syntax.

Fig. 12: Add Advanced Escalation Column to *Escalated Tickets* Screen Configuration

Note: To make the column available to users, but not visible by default, switch `IsVisible` key to 1.

- Click on the tick button on the right to save the setting.
- Deploy the modified system configuration.



The screenshot shows the 'Escalated Tickets (1 Ticket)' screen. At the top, there is a breadcrumb navigation 'Escalated Tickets (1 Ticket)' and a settings gear icon. Below that, there are icons for a document and a chart, and a status 'Unlocked' with a dropdown arrow. The main table has columns: Queue, Owner, Customer ID, First Response (Escalation reached, yes/no), Update Time (Escalation reached, yes/no), and Solution Time (Escalation reached, yes/r). The data row shows: Raw, agent-1 agent-1, customer-company-1, Yes, Yes.

Queue	Owner	Customer ID	First Response (Escalation reached, yes/no)	Update Time (Escalation reached, yes/no)	Solution Time (Escalation reached, yes/r)
Raw	agent-1 agent-1	customer-company-1	Yes	Yes	

Fig. 13: Advanced Escalation Columns in the *Escalated Tickets* Screen

The referenced advanced escalation column will now be visible by default in the *Escalated Tickets* screen, for all available escalation types.

To switch on any other supported advanced escalation column, change the column name to the following format:

```
EscalationType_ColumnIdentifier
```

Where:

ColumnIdentifier Any of supported column identifiers as listed in the *Possible Advanced Escalation Columns* section below.

For example, to switch on the `EscalationRemainingTime` column, use the following name:

```
EscalationType_EscalationRemainingTime
```

It is also possible to switch on the advanced escalation column for a particular escalation type only. To do this, change the column name to the following format:

```
Escalation_ID_ColumnIdentifier
```

Where:

ID Specific escalation type ID.

You can find this out by visiting the administrator interface, and editing the type in the *Escalation Types* module. It will be visible in the URL address field:

```
.../otrs/index.pl?Action=AdminEscalationType;Subaction=Edit;ID=1;
```

In the example above, the ID is 1.

ColumnIdentifier Any of supported column identifiers as listed in the *Possible Advanced Escalation Columns* section below.

For example, to switch on the `EscalationWorkingTime` column, but only for the escalation type with ID of 1, use the following name:

```
Escalation_1_EscalationWorkingTime
```

To change the default behavior of the general escalation columns which are available, but not visible by default, modify the column configuration in the following way:

```
Escalation_ID:
  isVisible: 0|1|2
```

Where:

ID Specific escalation type ID.

You can find this out by visiting the administrator interface, and editing the type in the *Escalation Types* module. It will be visible in the URL address field:

```
.../otrs/index.pl?Action=AdminEscalationType;Subaction=Edit;ID=1;
```

In the example above, the ID is 1.

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).

For example, to make the general column for the escalation type with ID of 1 visible by default, provide the following configuration:

```
Escalation_1:  
  IsVisible: 2
```

Possible Advanced Escalation Columns

For each configured escalation type the following 22 advanced escalation columns are calculated and available. You will find a complete list with explanations below.

EscalationDatetime Timestamp of the escalation date.

EscalationReached Yes/no value whether escalation time is reached.

EscalationTime Seconds until the escalation type is breached.

EscalationWorkingTime Seconds until the escalation type is breached with working calendar (only if calendar is defined).

EscalationRemainingTime Seconds until the escalation type is breached if suspended/stopped (only set if given).

EscalationRemainingWorkingTime Seconds until the escalation type is breached if suspended/stopped with working calendar (only if calendar is defined and if given).

NotifyDatetime Timestamp of notify start (only set if given).

NotifyTime Seconds until the notify start (only if given).

NotifyRemainingTime Seconds remaining until notify start if suspended/stopped (only set if given).

NotifyRemainingWorkingTime Seconds remaining until notify start if suspended/stopped with working calendar (only if calendar is defined).

SuspendLastTime Total amount of seconds that the timer was suspended last time.

SuspendLastWorkingTime Total amount of seconds that the timer was suspended last time with working calendars (only if calendar is defined).

SuspendTotalTime Total amount of seconds that the timer was suspended.

SuspendTotalWorkingTime Total amount of seconds that the timer was suspended with working calendars (only if calendar is defined).

RunningTotalTime Total amount of seconds that the timer was running (excluding suspend times).

RunningTotalWorkingTime Total amount of seconds that the timer was running (excluding suspend times) with working calendars.

RunningTotalVirtualTime Total amount of seconds that the timer was running with `BaseTime` as start date (excluding suspend times).

RunningTotalVirtualWorkingTime Total amount of seconds that the timer was running with `BaseTime` as start date (excluding suspend times) with working calendars (only if calendar is defined).

RunningLastTime Total amount of seconds that the timer was running last time (excluding suspend times).

RunningLastWorkingTime Total amount of seconds that the timer was running last time (excluding suspend times) with working calendars.

RunningLastVirtualTime Total amount of seconds that the timer was running last time with `BaseTime` as start date (excluding suspend times).

RunningLastVirtualWorkingTime Total amount of seconds that the timer was running last time with `BaseTime` as start date (excluding suspend times) with working calendars (only if calendar is defined).

2.1.4 Example Usage

This chapter describes how to add a first response time escalation.

Creating the Escalation Type

Go to the administration interface and choose *Escalation Type*. Click on the *Create New Escalation Type* button.

Fill in the needed fields. After submitting the form, you will be redirected to the edit screen for the new escalation type. You will see the *Start* tab for the new escalation type. Choose from the available conditions which one you want to be needed in order to start the escalation. Switch to another tab to set up more conditions.

In this example, we would enable the *The ticket has been created by a customer* setting and the *An agent message is not present*. Use the drop-down menu in order to choose whether or not an agent message has to be present.

Switch to the *Stop* tab in order to set up the escalation stop conditions. In this case, the escalation should be stopped if an agent sends a message, so we enable the *An agent sent a message* setting.

Creating the Escalation Type Bundle

Go to the administration interface and choose *Escalation Type Bundle*. Click on the *Create new escalation type bundle* button.

Fill in the needed fields. See the description on the previous pages in order to fill in *Execution order* correctly. After submitting the form, you will be redirected to the edit screen for the new bundle. You will now be able to add your previously created escalation type by choosing it from the *Add escalation type* drop-down menu. After you have added the escalation type, you can set up its parameters. Set up a time span (like *1*) and a time unit (like *Hour(s)*). Save the screen.

Assigning the New Bundle to an SLA

In order to get the new escalation type to work, you need to assign it to an existing SLA. Go to the SLA management screen for an existing SLA and select your bundle from the list of available bundles. Save the screen.

Conclusion

The new escalation type in connection with the bundle and SLA assignment will cause new tickets which are being created by customers to escalate after one hour if no agent has responded.

2.2 Agent Interface

For assigning an advanced escalation to a ticket, this one should contain a valid customer ID and valid SLA, so the advanced escalation for the ticket will be assigned based on previous configuration. Advanced escalations currently does not work for unknown customers.

It is possible to show advanced escalations columns in ticket lists, ticket list widgets and statistics. The advanced escalations data is calculated on-the-fly and could have a huge negative performance impact on the whole system.

Note: Escalation automatism is not supported. Use the generic agent or generic interface to trigger escalation events.

2.2.1 Tickets

After installation of the package a new *Advanced Escalations* widget is displayed in the ticket detail view.

Display Escalations in a Widget

All advanced escalations that are associated to a ticket will be shown in *Advanced Escalations* widget in the ticket detail view.

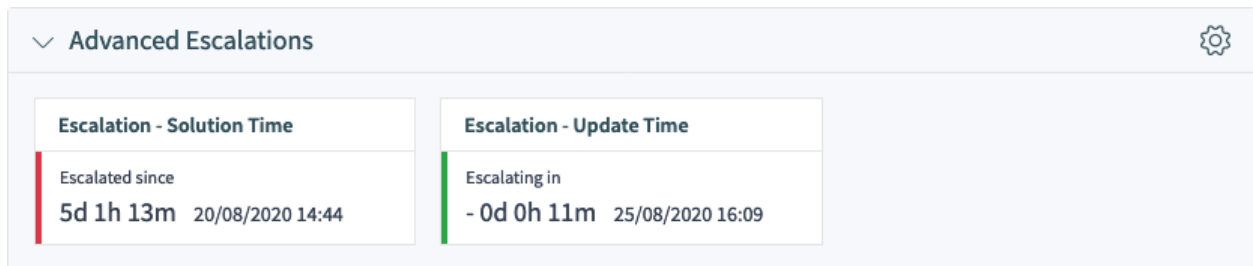


Fig. 14: Advanced Escalations Widget in Ticket Detail View

The escalations are displayed as cards with a colored bar, indicating the current escalation state for better visibility.

- Green bar: the escalation time has not been reached.
- Orange bar: the escalation notification time has been reached.
- Red bar: the escalation time has been reached.
- Grey bar: the escalation is suspended.

Debug Escalations

It is not easy to understand how advanced escalations work. It cuts the ticket history entries into snapshots to calculate the escalation. Each snapshot will contain the current state of the ticket with all related data of the current time of the snapshot. A snapshot is normally about 5 seconds of the ticket history.

For more information about a single escalation and how it is calculated you can take a look at the ticket history. If your system has a customer ID and a SLA with related escalation bundles and types then you will be able to debug these, based on the ticket history of the current ticket.

To get access to the debug information, click on *View History* ticket action in the ticket detail view.

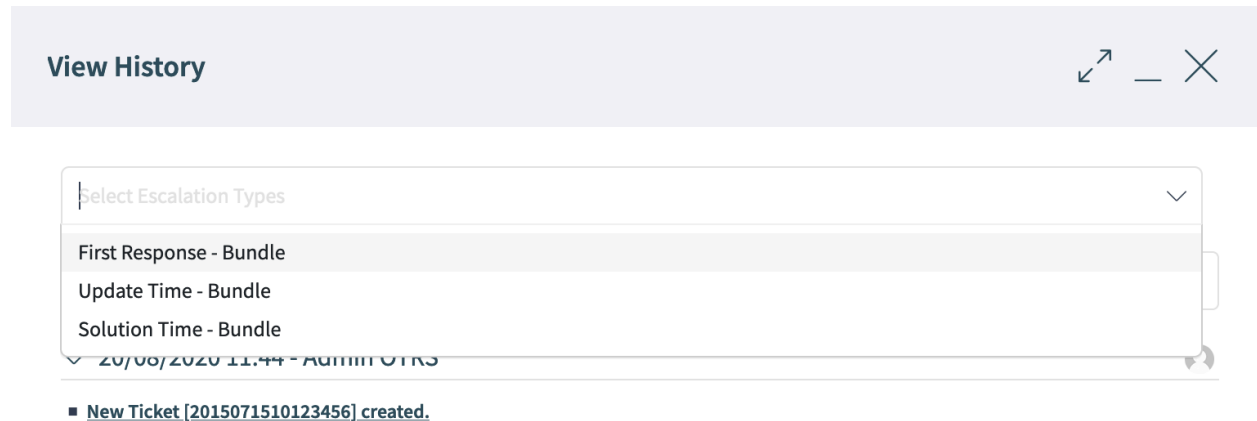


Fig. 15: Escalation Debug Mode – Select Escalation Type

After choosing an escalation type from the list, you will get a detailed information about the parameters, attribute values and conditions, right below relevant ticket history entry.

Escalation State

The first block contains the state of the current escalation type. In the example it will calculate the start condition of the escalation type and it was successful (success = green, red = no change of the state). Basically, for each snapshot it is possible that the escalation will calculate multiple state if it does not match to the first.

The order of the calculated states is based on the running, suspended or not running state of the escalation. Here is a short technical overview:

```
my %EscalationStatesMachine = (
    NotRunning => [ 'Start' ],
    Running    => [ 'Suspend', 'Restart', 'Stop' ],
    Suspended  => [ 'Resume', 'Stop' ],
);
```

So if the escalation is not running then it will only try to move to the *Start* state of the escalation. If it is suspended then only *Resume* or *Stop* are possible as next states.

21/08/2020 16:10 - agent-1 agent-1



- Changed customer to "CustomerID=customer-company-1;CustomerUser=customer-1;"

Debug - Start - First Response - Bundle

Attribute Sets

Attribute	Set	Success	Fail
	1	Agent Message Present,Is_StateType,WasNever_StateType	

Attributes

Attribute	Currently	Previous	Was
Priority	3 normal	3 normal	
Queue	Raw	Raw	
StateType	new	new	
State	new	new	
TicketType	Unclassified	Unclassified	

Fixed Filters

Attribute	Filters
	Agent Message Present
	Agent Message Sent
	Created By Customer
	Customer Message Present
	Customer Message Sent

Fig. 16: Escalation Debug Mode – Debug Snapshot of Escalation Type

> Debug - Start - Solution Time - Bundle

Fig. 17: Escalation Debug Mode – Escalation State

Escalation Attribute Sets

The block with the attribute sets displays if the attribute sets of the escalation type for the given escalation state is matching for the current snapshot.

Attribute Sets			
Attribute	Set	Success	Fail
●	1		Is_StateType
●	2	Agent Message Sent	

Fig. 18: Escalation Debug Mode – Escalation Attribute Sets

There are 4 columns:

- **Attribute:** The circle will be displayed green if it matches or red if not.
- **Set:** This column contains the number of the attribute set in the escalation type. The number will be displayed green if it matches or red if not.
- **Success and Fail:** These columns show the matched conditions of an attribute set. At this point it will use internal names for matching the attributes. Here are some examples:
 - `Agent Message Present`: A message from an agent is present.
 - `Agent Message Sent`: An agent sent a message.
 - `Customer Message Present`: A message from a customer is present.
 - `Customer Message Sent`: A customer sent a message.
 - For ticket attributes it will show the internal condition:

```
Is      => "equals now",
IsNot   => "doesn't equal now",
Was     => "equaled",
WasNever => "did never equal",
ChangedTo => "changed to",
IsPrev  => "equals previous",
IsNotPrev => "doesn't equal previous",
```

- `Is_StateType`: The ticket attribute condition `StateType` of the attribute set of the current snapshot matched if green or not matched if red.

Escalation Attributes

The block with the attributes of the snapshot will contain all attributes which will be used for checking the condition of the escalation ticket attributes.

There are 4 columns:

- **Attribute:** This column contains the ticket attribute.
- **Currently:** This column contains the current value of the snapshot. This value will be used for the calculation of *equals now*, *doesn't equal now* and *changed to* conditions.

Attributes			
Attribute	Currently	Previous	Was
Priority	3 normal	3 normal	
Queue	Raw	Raw	
StateType	open	new	new
State	open	new	new
TicketType	Unclassified	Unclassified	

Fig. 19: Escalation Debug Mode – Escalation Attributes

- **Previous:** This column contains the value of the previous snapshot. This value will be used for the calculation of *equals previous*, *doesn't equal previous* and *changed to* conditions.
- **Was:** This column contains the value of all previous snapshot. This value will be used for the calculation of *did never equal* and *equaled* conditions.

Escalation Filters

The filter values are relating to the agent and customer message send or present attributes which are set in the conditions. If a filter is green then the filter is matched and red if a filter is not matched.





Fixed Filters	
Attribute	Filters
	Agent Message Present
	Agent Message Sent
	Customer Message Present
	Customer Message Sent

Fig. 20: Escalation Debug Mode – Escalation Filters

- **Agent Message Present:** A message from an agent is present.
- **Agent Message Sent:** An agent sent a message.
- **Customer Message Present:** A message from a customer is present.
- **Customer Message Sent:** A customer sent a message.

2.2.2 Statistics and Reports

The escalation information can be displayed in reports and statistics using the new statistics object modules.

Advanced Escalations Data in Statistics

By installing the advanced escalations package, you have the possibility to use the new statistics object modules:

TicketAccountedTimeEscalation A new matrix statistic which only contains tickets with accounted time. It is a copy of the `TicketAccountedTime` statistic out of the **OTRS** framework, but you are able to show advanced escalations data and to filter the data by ticket attributes as well as by advanced escalations.

TicketEscalation A new matrix statistic which shows configured columns on X- and Y-axis. It is a copy of the ticket matrix statistic out of the **OTRS** framework, but you are able to show advanced escalations data and to filter the data by ticket attributes as well as by advanced escalations.

TicketListEscalation A new statistic which shows configured columns (advanced escalations data, too) on X-axis. You can specify a column which is used in an order by clause on Y-axis. You are able to filter the data by ticket attributes as well as by advanced escalations columns.

TicketSolutionResponseTimeEscalation A new matrix statistic which only contains closed tickets. It is a copy of the default `TicketSolutionResponseTimeEscalation` statistic out of the **OTRS** framework, but you are able to select advanced escalations columns in the *Elevation by* field. As already mentioned in the other new statistics, you can filter the data by advanced escalations columns.

Filtering

In general, there are the following types of advanced escalations columns:

Yes/no columns For this type of columns (e.g. `EscalationReached`), a drop-down field is displayed to select the wanted values. This type is used on X- and Y-axis and as filter.

Datetime columns For this type of columns (e.g. `EscalationDatetime`), a complex field with the selection of an absolute or a relative time range is shown. This type is used on X- and Y-axis and as filter.

Note: Please note that in the `TicketListEscalation` statistic these columns are not available on the Y-axis.

Time columns (seconds) For this type of columns (e.g. `RunningTotalTime`), a normal input field is shown. You can use these columns only as filter as *greater or equal* and/or *smaller or equal*.

Entered values are treated as minutes. For example if you enter 30 it is converted to $30 \times 60 = 1800$ seconds automatically.

2.3 External Interface

This package has no external interface.

Advanced Generic Agent

This feature add-on enables the OTRS administrators to define actions for certain tickets that are carried out automatically in the agent interface. It is therefore possible to, for example, close tickets automatically using predetermined attributes (*pending*, *escalated*, etc.) or send notifications to the owner of tickets escalating soon.

The advanced generic agent now makes it possible to also assign a time value in the form of a time span instead of a certain point in time, i.e. *+ 2 weeks* or *-3 months*, to the dynamic field. Employees are therefore able to specify resubmission periods or organize recurring communication processes with special customers easily and at regular intervals.

Benefits

- Makes creating and executing automated ticket actions much easier.
- Faster processing of projects and tasks with time limits.
- Prevents process errors and processes from being forgotten.
- Saves work through automation.

Target Groups

- IT service management
- Customer service/support
- Sales
- Universities
- Government

Available in Service Package

- GOLD

Package Name in OTRS Package Manager

- OTRSAdvancedGenericAgent

3.1 Administrator Interface

This package has no dedicated administrator interface, only extends the existing screens.

3.1.1 Generic Agent

This package extends the ticket actions of the *Generic Agent* module of the administrator interface. Dynamic fields of type date or date and time will be extended with the functionality to set absolute, relative or no date for the dynamic field.

To set relative dates:

1. Open the *Generic Agent* screen in administrator interface.
2. Expand the tab *Update/Add Ticket Attributes*.
3. Set relative dates for date or date and time dynamic fields.

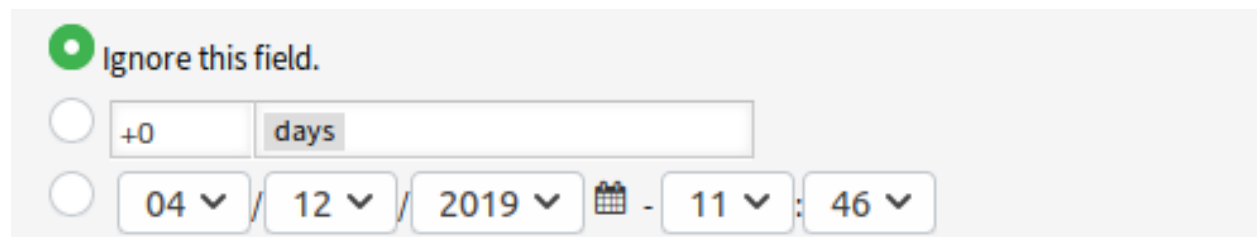
The screenshot shows a user interface for setting a date field. At the top, there is a radio button that is selected, labeled "Ignore this field." Below this, there are two more radio buttons. The first one is unselected and is followed by a text input field containing "+0" and a dropdown menu with "days" selected. The second radio button is also unselected and is followed by a date and time picker. The date picker shows "04" for the month, "12" for the day, and "2019" for the year, with a calendar icon. To the right of the date is a minus sign, followed by a time picker showing "11" for the hour and "46" for the minutes, both with dropdown arrows.

Fig. 1: Date Field in Generic Agent Screen

The values can be positive like *+5 days* or negative like *-10 hours* based on the current time.

3.2 Agent Interface

This package has no agent interface.

3.3 External Interface

This package has no external interface.

Advanced Ticket Article Forward

With this feature add-on, ticket articles can now be sent as email attachments. This makes communication and information transfer faster and more efficient. It also helps to avoid errors caused by incorrect documents being sent.

With this new option, relevant files no longer have to be located and downloaded before being added to a new email. Service staff can select individual or multiple ticket articles and forward them directly to external recipients: Any attachment related to the selected article will be sent along with the new email.

The feature add-on applies to any type of ticket article.

Benefits

- Saves time and effort. No more searching for and downloading relevant files.
- Enables faster communication.
- Relieves the agents.
- Reduces sources of error.

Target Groups

- Customer service
- IT service
- Project management
- Document management
- Facility management
- Human resources
- Logistics

Available in Service Package

- SILVER

Package Name in OTRS Package Manager

- OTRSAdvancedTicketArticleForward

4.1 Administrator Interface

This package has no administrator interface.

The advanced article forward functionality can be disabled in the system configuration.

4.2 Agent Interface

This package adds the bulk forward functionality for articles and the possibility to display an advanced article forward icon in the article widgets, regardless of used communication channel, sender type and visible to customer setting.

4.2.1 Ticket Detail View

The feature makes it possible for the agents to select multiple articles from one ticket and forward them at the same time instead of forwarding only a single article. A checkbox is displayed next to each article in the *Communication Stream* and in the *Communication Compact* widgets and a mail icon is displayed in the header of the widget. Clicking on this icon opens the *Forward via Email* action and attaches all selected articles as attachments.

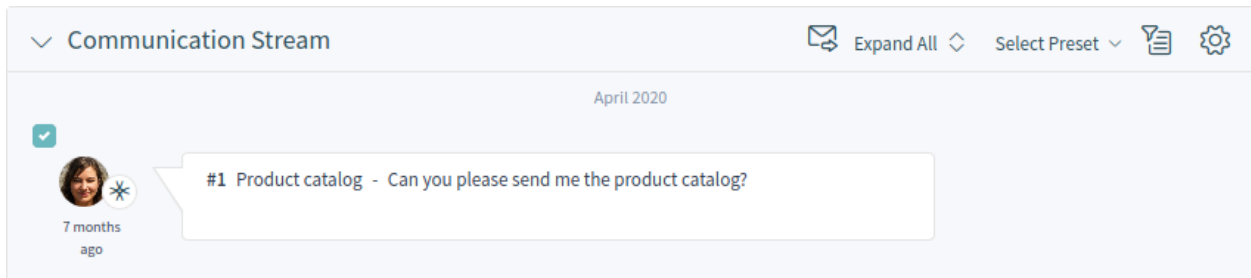


Fig. 1: Advanced Article Forward in Communication Stream Widget

It is possible to set in the system configuration that the forward icon is displayed for every ticket articles. In OTRS usually notes, notifications and email notifications are not forwardable.

By using the advanced article forward functionality it is also possible to set all selected articles as seen.

Use case example:

1. The agent opens the ticket detail view to forward multiple articles. With the checkboxes in the article list, the relevant articles can be selected or all articles with the checkbox in the header row. The advanced article forward icon is disabled, if no article in the article list is selected. The agent can also forward a note if this possibility is activated in the system configuration settings.
2. The advanced article forward icon will be activated, if the agent selects at least one article.
3. In the *Forward via Email* action the selected articles are added as attachments.
4. If the feature is activated in the system configuration setting, the agent can set all forwarded articles as seen with sending the forward mail.

4.3 External Interface

This package has no external interface.

Agent Email Interface

This feature add-on makes it possible for an agent to work with tickets via emails without having to use the web agent interface of **OTRS**.

In order to work with this module an agent could just reply to any ticket notification email by including one or many commands between two special command tags, like this:

```
<OTRS_CMD> send </OTRS_CMD>
```

or

```
<OTRS_CMD> send, nocc </OTRS_CMD>
```

Commands can be combined (comma separated) if it makes sense to combine them.

The following commands exist:

- `send` – Sends an email to the customer, including the cc and bcc recipients (`send` also locks the ticket).
- `nocc` – If used together with `send`, the cc and bcc recipients will be excluded.
- `lock` – Locks the ticket.
- `unlock` – Unlocks the ticket.
- `get` – Gets the last customer article and sends it to the agent.
- `note` – Adds an internal note to the ticket.
- `close` – Closes the ticket (also unlocks the ticket).

Note: The package restricts the functionality of this feature to agents known to the system, but it does not use advanced security mechanisms such as digital signatures and the like.

Benefits

- Saves time because frequent logging in and out is not necessary.

- Increase flexibility because you do not have to log in to the system.

Target Groups

- Can be used in all areas where agents are not logged in permanently to the **OTRS** agent interface.

Available in Service Package

- GOLD

Package Name in OTRS Package Manager

- OTRSAgentEmailInterface

5.1 Administrator Interface

This package has no administrator interface.

5.1.1 System Configuration

The agent interface address should be set to the same as in setting *Core* → *Email* → *NotificationSenderEmail*.

To check if the addresses are the same:

1. Go to *System Configuration* screen.
2. Select *OTRS* in the *Navigation* widget.
3. Navigate to *Core* → *Email* in the navigation tree and search for setting `NotificationSenderEmail`.
4. Change the email address if needed, and remember it.
5. Select *OTRSAgentEmailInterface* in the *Navigation* widget.
6. Navigate to *Core* → *Email* → *PostMaster* in the navigation tree.
7. Change the value of key `AgentInterfaceAddress` to the same as set in step 3.

5.2 Agent Interface

This package has no agent interface.

5.2.1 Usage

Reply any ticket notification and insert for example the following tag anywhere in the body of the mail:

```
<OTRS_CMD> close </OTRS_CMD>
```

Switch to the ticket from the ticket notification and compare the ticket state. The ticket should have the state *closed*.

Note: The package restricts the functionality of this feature to agents known to the system, but it does not use advanced security mechanisms such as digital signatures and the like.

5.3 External Interface

This package has no external interface.

Automated FAQ Ticket Creator

With the help of this feature add-on, it is possible to create time-controlled tickets via knowledge base articles. This can refer to the knowledge base article itself or to a specific task that is related to the knowledge base article topic. This is particularly useful in situations, such as maintenance work, where routine activities take place at large intervals. The automatically generated ticket becomes a *To do* for the employee. And, through the knowledge base article documentation, everyone knows the steps of the task.

Example An employee creates a knowledge base article for a specific maintenance task. The feature add-on will now automatically create an associated ticket after a specified period of time, reminding an employee to either renew the knowledge base article or to perform the activity described in the knowledge base article. For new employees who do not have experience, the knowledge base article makes it possible for them to perform the activity too, because the steps are clearly described in the documentation.

Benefits

- Keep knowledge base articles up-to-date with reminders about necessary update.
- Less frequently opened *resubmission tickets*.
- Functionality can be used for one-time activities that include an end date.
- Even complex tasks can be done by new or untrained employees.

Target Groups

- IT service management
- Facility management
- Service providers
- Companies with recurring activities, such as maintenance work
- Companies with a large number of employees

Available in Service Package

- TITANIUM

Package Name in OTRS Package Manager

- OTRSAutomatedFAQTicketCreator

6.1 Administrator Interface

This chapter describes the new features that are available in the administrator interface after installation of the package.

6.1.1 Processes & Automation

After installation of the package some new dynamic fields are added to the system.

Dynamic Fields

After installation of the package some new dynamic fields are added to the system. The dynamic field management screen is available in the *Dynamic Fields* module of the *Processes & Automation* group.

New Dynamic Fields

This packages provides new dynamic fields and an OTRS daemon cron job to create new tickets with configured values based on knowledge base articles.

OTRSAutomatedFAQTicketCreatorStartTime This dynamic field defines the start time for the ticket creation. This is a date/time field.

OTRSAutomatedFAQTicketCreatorEndTime This dynamic field defines the end time for the ticket creation. This is a date/time field.

OTRSAutomatedFAQTicketCreatorFrequency This dynamic field defines the frequency for the ticket creation. Possible values are: *Daily, Weekly, Monthly, Quarterly, Yearly*.

To create a ticket for each month choose the frequency *Monthly*.

OTRSAutomatedFAQTicketCreatorRepeatTimes This dynamic field defines the repeat times for the ticket creation. Possible values are: 1-20.

If you choose the frequency *Monthly* and a repeat time of 2 then the ticket will be created each second month: January, March, May, July, etc.

OTRSAutomatedFAQTicketCreatorRepeatOnDays This dynamic field defines the repeat on days for the ticket creation. Possible values are: *Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday*.

If you choose the frequency *Daily*, a repeat time of 2 and *Monday to Friday* as repeat on days then the ticket will be created each second day in the range between Monday to Friday: Monday, Wednesday Friday, Monday, etc.

OTRSAutomatedFAQTicketCreatorLastExecution This dynamic field defines the last execution time for the ticket creation. This is a date/time field.

OTRSAutomatedFAQTicketCreatorSubject This dynamic field defines the article subject for the ticket creation. Possible value is: *[Article title]*, for example: *This is the subject of the article*.

See also:

To define a default value for this dynamic field, you can also set a value to the following system configuration option:



















Dynamic Fields List ⚙️						
26-43 of 43 - Page: 12						
NAME	LABEL	ORDER	TYPE	OBJECT	VALIDITY	DELETE
OTRSAutomatedFAQTicketCreatorStartTime	Ticket create start time	26	Date / Time	FAQ	valid	
OTRSAutomatedFAQTicketCreatorEndTime	Ticket create end time	27	Date / Time	FAQ	valid	
OTRSAutomatedFAQTicketCreatorFrequency	Ticket create frequency	28	Dropdown	FAQ	valid	
OTRSAutomatedFAQTicketCreatorRepeatTimes	Ticket create repeat times	29	Dropdown	FAQ	valid	
OTRSAutomatedFAQTicketCreatorRepeatOnDays	Ticket create repeat on days	30	Multiselect	FAQ	valid	
OTRSAutomatedFAQTicketCreatorLastExecution	Ticket create last execution	31	Date / Time	FAQ	valid	
OTRSAutomatedFAQTicketCreatorSubject	Ticket create subject	32	Text	FAQ	valid	
OTRSAutomatedFAQTicketCreatorTitle	Ticket create title	33	Text	FAQ	valid	
OTRSAutomatedFAQTicketCreatorCustomerUser	Ticket create customer user	34	Dropdown	FAQ	valid	
OTRSAutomatedFAQTicketCreatorOwner	Ticket create owner	35	Dropdown	FAQ	valid	
OTRSAutomatedFAQTicketCreatorResponsible	Ticket create responsible	36	Dropdown	FAQ	valid	
OTRSAutomatedFAQTicketCreatorPriority	Ticket create priority	37	Dropdown	FAQ	valid	
OTRSAutomatedFAQTicketCreatorState	Ticket create state	38	Dropdown	FAQ	valid	
OTRSAutomatedFAQTicketCreatorService	Ticket create service	39	Dropdown	FAQ	valid	
OTRSAutomatedFAQTicketCreatorBody	Ticket create body	40	Multiselect	FAQ	valid	
OTRSAutomatedFAQTicketCreatorSLA	Ticket create SLA	41	Dropdown	FAQ	valid	
OTRSAutomatedFAQTicketCreatorType	Ticket create ticket type	42	Dropdown	FAQ	valid	
OTRSAutomatedFAQTicketCreatorQueue	Ticket create queue	43	Dropdown	FAQ	valid	

Fig. 1: Dynamic Field Management Screen

- `OTRSAutomatedFAQTicketCreator::DefaultValues###Subject`

OTRSAutomatedFAQTicketCreatorTitle This dynamic field defines the ticket title for the ticket creation. Possible value is: *[Ticket title]*, for example: *This is the title of the ticket*.

See also:

To define a default value for this dynamic field, you can also set a value to the following system configuration option:

- `OTRSAutomatedFAQTicketCreator::DefaultValues###Title`

OTRSAutomatedFAQTicketCreatorCustomerUser This dynamic field defines the ticket customer user for the ticket creation. Possible values are: *[Customer user login]* → *[Customer user full name]*, for example: *customer-1* → *John Doe*.

See also:

To define a default value for this dynamic field, you can also set a value to the following system configuration option:

- `OTRSAutomatedFAQTicketCreator::DefaultValues###CustomerUser`

OTRSAutomatedFAQTicketCreatorOwner This dynamic field defines the ticket owner for the ticket creation. Possible values are: *[Owner user login]* → *[Owner full name]*, for example: *root@localhost* → *Admin OTRS*.

See also:

To define a default value for this dynamic field, you can also set a value to the following system configuration option:

- `OTRSAutomatedFAQTicketCreator::DefaultValues###Owner`

OTRSAutomatedFAQTicketCreatorResponsible This dynamic field defines the ticket responsible for the ticket creation. Possible values are: *[Responsible user login]* → *[Responsible full name]*, for example: *root@localhost* → *Admin OTRS*.

See also:

To define a default value for this dynamic field, you can also set a value to the following system configuration option:

- `OTRSAutomatedFAQTicketCreator::DefaultValues###Responsible`

OTRSAutomatedFAQTicketCreatorPriority This dynamic field defines the priority for the ticket creation. Possible values are: *[Priority Name]* → *[Priority Name]*, for example: *very low* → *1 very low*.

See also:

To define a default value for this dynamic field, you can also set a value to the following system configuration option:

- `OTRSAutomatedFAQTicketCreator::DefaultValues###Priority`

OTRSAutomatedFAQTicketCreatorState This dynamic field defines the ticket state for the ticket creation. Possible values are: *[State name]* → *[State name]*, for example: *pending reminder* → *pending reminder*.

See also:

To define a default value for this dynamic field, you can also set a value to the following system configuration option:

- `OTRSAutomatedFAQTicketCreator::DefaultValues###State`

OTRSAutomatedFAQTicketCreatorService This dynamic field defines the ticket service for the ticket creation. Possible values are: *[Service name] → [Service name]*, for example: *1st Level Service → 1st Level Service*.

See also:

To define a default value for this dynamic field, you can also set a value to the following system configuration option:

- `OTRSAutomatedFAQTicketCreator::DefaultValues###Service`

OTRSAutomatedFAQTicketCreatorBody This dynamic field defines the article body for the ticket creation. Possible values are: *[Knowledge base article field index] → [Knowledge base article field caption]*, for example: *1 → Symptom*.

See also:

To define a default value for this dynamic field, you can also set a value to the following system configuration option:

- `OTRSAutomatedFAQTicketCreator::DefaultValues###Body`

OTRSAutomatedFAQTicketCreatorSLA This dynamic field defines the ticket SLA for the ticket creation. Possible values are: *[SLA name] → [SLA name]*, for example: *1st Level SLA → 1st Level SLA*.

See also:

To define a default value for this dynamic field, you can also set a value to the following system configuration option:

- `OTRSAutomatedFAQTicketCreator::DefaultValues###SLA`

OTRSAutomatedFAQTicketCreatorType This dynamic field defines the ticket type for the ticket creation. Possible values are: *[Ticket type name] → [Ticket type name]*, for example: *Unclassified → Unclassified*.

See also:

To define a default value for this dynamic field, you can also set a value to the following system configuration option:

- `OTRSAutomatedFAQTicketCreator::DefaultValues###Type`

OTRSAutomatedFAQTicketCreatorQueue This dynamic field defines the queue for the ticket creation. Possible values are: *[Queue name] → [Queue name]*, for example: *Raw → Raw*.

See also:

To define a default value for this dynamic field, you can also set a value to the following system configuration option:

- `OTRSAutomatedFAQTicketCreator::DefaultValues###Queue`

To add new values:

1. Choose the dynamic field you would like to change the values for.
2. Add a new value in the *Field Settings* widget.
3. Click on the *Save* button to save the dynamic field.

System Configuration

Use the following system configuration option to copy dynamic field values of the knowledge base article to the ticket:

- `OTRSAutomatedFAQTicketCreator::Core::DynamicFieldMapping###DynamicField`

To define a default value for this dynamic field, you can also set a value to the following system configuration option:

- `OTRSAutomatedFAQTicketCreator::DefaultValues###DynamicField`

Usage

The following use case example will show how to edit and add a customer user.

To edit field values of a dynamic field:

1. Go to the *Dynamic Fields* module of the administrator interface.
2. Choose the dynamic field you like to change the values for.
3. Add a new value in the *Field Settings* widget.
4. Click on the *Save* or *Save and finish* button to save the dynamic field.

For example copy the login name, first name and last name of the customer user to the field.

Fig. 2: Edit Dynamic Field Value

To edit default field values of a dynamic field:

1. Go to *System Configuration* screen.
2. Select `OTRSAutomatedFAQTicketCreator` in the *Navigation* widget.
3. Navigate to `Core` → `OTRSAutomatedFAQTicketCreator` → `DefaultValues` in the navigation tree.
4. Add the default value for customer user to setting `OTRSAutomatedFAQTicketCreator::DefaultValues###CustomerUser`

Fig. 3: System Configuration – Default Value

To copy a field value of a knowledge base article dynamic field into the new ticket dynamic field:

1. Go to *System Configuration* screen.
2. Select *OTRSAutomatedFAQTicketCreator* in the *Navigation* widget.
3. Navigate to *Core* → *OTRSAutomatedFAQTicketCreator* in the navigation tree.
4. Search for setting `OTRSAutomatedFAQTicketCreator::Core::DynamicFieldMapping###DynamicField`.
5. Set mapping for a dynamic field. You need to use the ticket field as key and the knowledge base article field as value.

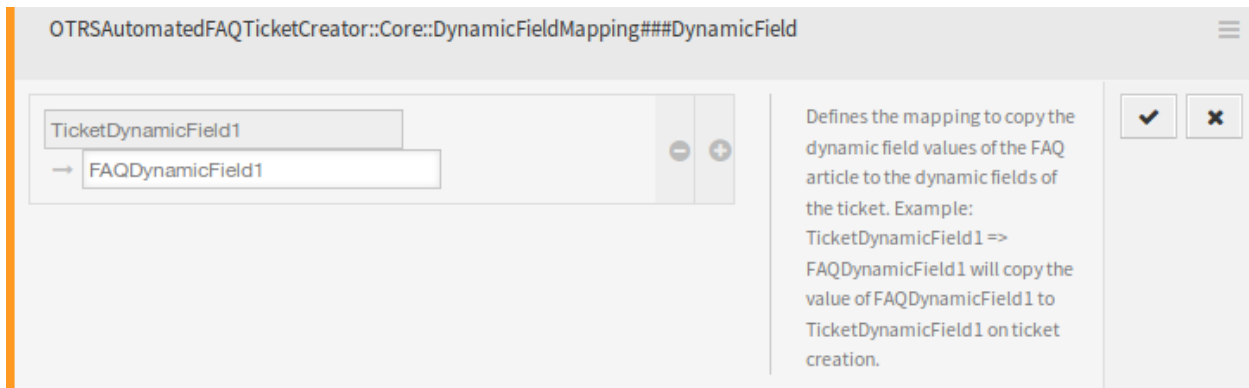


Fig. 4: System Configuration – Copy Value

Note: It is only possible to map dynamic fields of the same type.

To set default field values of a dynamic field for the ticket created dynamic fields:

1. Go to *System Configuration* screen.
2. Select *OTRSAutomatedFAQTicketCreator* in the *Navigation* widget.
3. Navigate to *Core* → *OTRSAutomatedFAQTicketCreator* → *DefaultValues* in the navigation tree.
4. Search for setting `OTRSAutomatedFAQTicketCreator::DefaultValues###DynamicField`.
5. Set some default values, for example:
 - **Text:** `ExampleTicketDynamicField1Text` → *hallo*
 - **Multiselect:** `ExampleTicketDynamicField1Multiselect` → *Value1;Value2;Value3*
 - **Date:** `ExampleTicketDynamicField1Date` → *2014-03-03*
 - **Date/Time:** `ExampleTicketDynamicField1DateTime` → *2014-03-03 10:00:00*
 - **Checkbox:** `ExampleTicketDynamicField1Checkbox` → *1*

6.2 Agent Interface

This package has no dedicated agent interface, but the *Create Knowledge Base Article* and the *Update Knowledge Base Article* screens are updated with new dynamic fields.

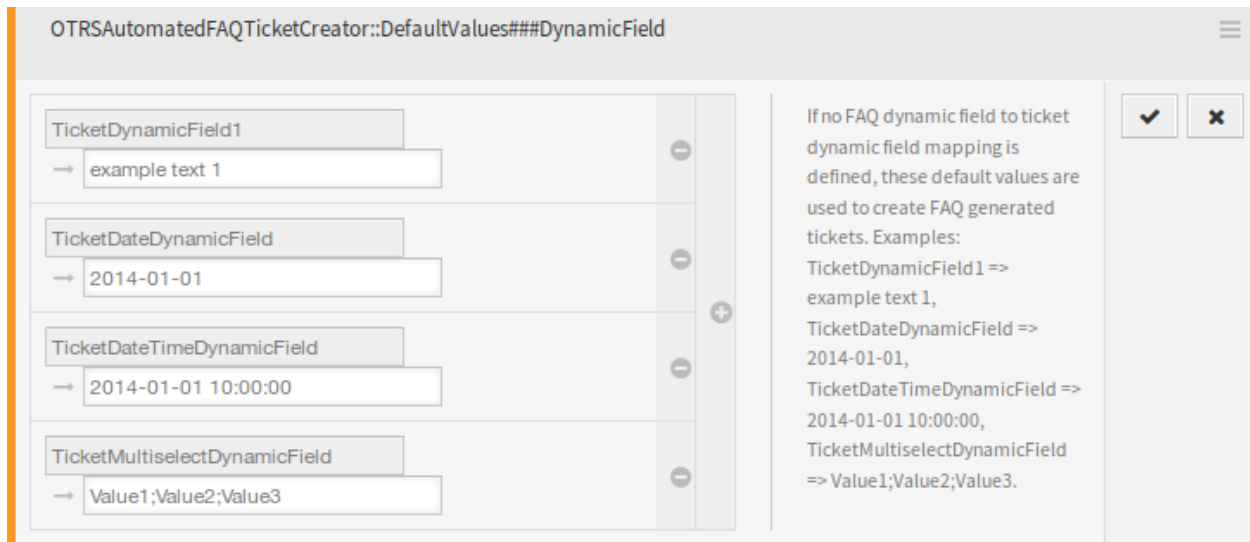


Fig. 5: System Configuration – Default Field Value

6.2.1 Knowledge Base Articles

After installation of the package some new dynamic field are added to the *Create Knowledge Base Article* and the *Update Knowledge Base Article* screens.

It is possible to define the ticket fields to be auto-populated and the ones that has to be filled manually.

By default only certain dynamic fields are shown in the knowledge base article creation and edit screens, those that controls the frequency of the ticket creation but not ones that set values for the ticket, instead the ticket uses the default values provided in the system configuration settings described in *Dynamic Fields* chapter of the administrator interface.

See also:

To show the dynamic fields that set ticket values:

1. Go to *System Configuration* screen.
2. Search for settings `Forms###AgentFrontend::KnowledgeBaseArticleCreate::Properties` and/or `Forms###AgentFrontend::KnowledgeBaseArticleUpdate::Properties`.
3. Update their `Hidden` values from `1` to `0` as any other dynamic field.

Use Case Example

The following example assumes that all knowledge base article dynamic fields that set ticket values are activated for the *Create Knowledge Base Article* screen and the configuration of each dynamic field is set with valid values for the system.

To create a knowledge base article which will be triggered for the ticket creation:

1. Create a new knowledge base article.
2. Set `2020-09-15 10:00:00` for *Ticket create start time*.
3. Set `2021-09-15 10:00:00` for *Ticket create end time*.
4. Set `Daily` for *Ticket create frequency*.

5. Set *01* for *Ticket create repeat times*.
6. Select *Monday* and *Friday* for *Ticket create repeat on days*.
7. Fill the following fields to define the values for the ticket which will be created:
 - Ticket create customer user
 - Ticket create owner
 - Ticket create responsible
 - Ticket create ticket type
 - Ticket create queue
 - Ticket create priority
 - Ticket create service
 - Ticket create SLA
 - Ticket create state

✓ Automated Knowledge Base Article Ticket Creator Settings

Ticket create start time

09/15/2020 - 10:00:00 📅

Ticket create end time

09/15/2021 - 10:00:00 📅

* Ticket create frequency

Daily ▼

* Ticket create repeat times

01 ▼

Ticket create repeat on days

Friday × Monday × × ▼

Fig. 6: Create Knowledge Base Article

Now every Monday and Friday in the range of 2020-2021 years the OTRS daemon cron job will create a ticket for this knowledge base article.

6.3 External Interface

This package has no external interface.

Automatic Start of Processes

Recurring tasks can be digitized particularly well. Creating process tickets on a schedule relieves your agents, and even your customers, of the obligation to think about the tasks. Once set up, steps can be automated and begun without manual intervention. Documentation of task fulfillment develops naturally as the ticket moves through the process.

Benefits

- Automated processes save time.
- Tasks are completed efficiently.
- Performance is consistent and errors are reduced.
- The right tasks are completed at the right times.

Target Groups

- Complaint management
- Customer service
- Facility management
- Finance and accounting
- IT service
- Security management
- Service providers

Available in Service Package

- GOLD

Package Name in OTRS Package Manager

- OTRSAutomaticStartOfProcesses

7.1 Administrator Interface

This package implements a scheduled automatic creation of process tickets.

7.1.1 Processes & Automation

After installation of the package a new *Start Event* element is added to process management.

Process Management

After installation of the package a new *Start Event* element will be available in the accordion widget of the process management. Processes can now contain a normal start point and a scheduled start point. The schedule of the start event can be configured via an interface.

Create Start Events

To be able to create a start event for the process management, create a process or modify an existing one.

Click on the *Start Events* item in the *Available Process Elements* widget in the left sidebar. This action will expand the *Start Events* options and will collapse all others doing an accordion like effect. Click on the *Create New Start Event* button.

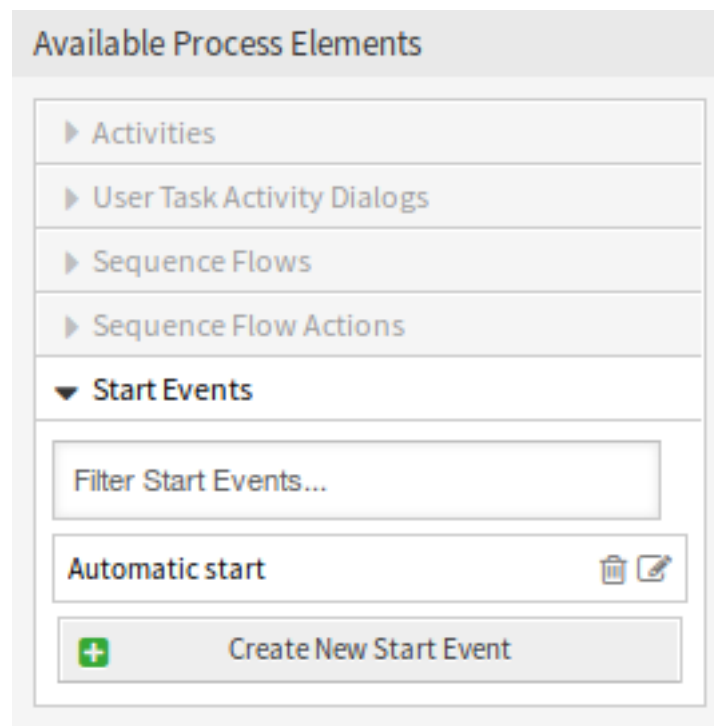


Fig. 1: Start Events

In the opened popup screen fill in the *Start Event Name* as well as set the schedule times.

Start event name A name of the scheduled based start point.

Schedule settings Cron settings of the scheduled start point.

Create new start event
Cancel & close

▼ Start event

★ Start event name:

▼ Schedule settings

SCHEDULE MINUTES	SCHEDULE HOURS	SCHEDULE DAYS
<input type="text"/>	<input type="text"/>	<input type="text"/>

Currently this start event will not run automatically.
To enable automatic execution select at least one value from minutes, hours and days!

Fig. 2: Create New Start Event Screen

To edit an already defined start event, just click on its name in the *Available Process Elements* widget.

Edit start event "Automatic start"
Cancel & close

▼ Start event

★ Start event name:

▼ Schedule settings

SCHEDULE MINUTES	SCHEDULE HOURS	SCHEDULE DAYS
<input type="text" value="00"/>	<input type="text" value="00"/>	<input type="text" value="Sun"/> <input type="text" value="Mon"/> and 5 more...

Fig. 3: Edit Start Event Screen

Add Start Event to Canvas

To add a configured start event to the process canvas screen a free activity (not connected to other activities) is necessary. The scheduled based start event can not be connected directly to the normal start point of the process. In addition, the timer start event is not moveable. It is absolutely required to connect the scheduled based start event to a free activity. This first scheduled based activity can then be connected with a normal activity.

The initial activity of the timer start event can be any type of activity. In case of a user task activity, the process ticket is created and the process stays in this activity.

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.

7.2 Agent Interface

This package has no agent interface.

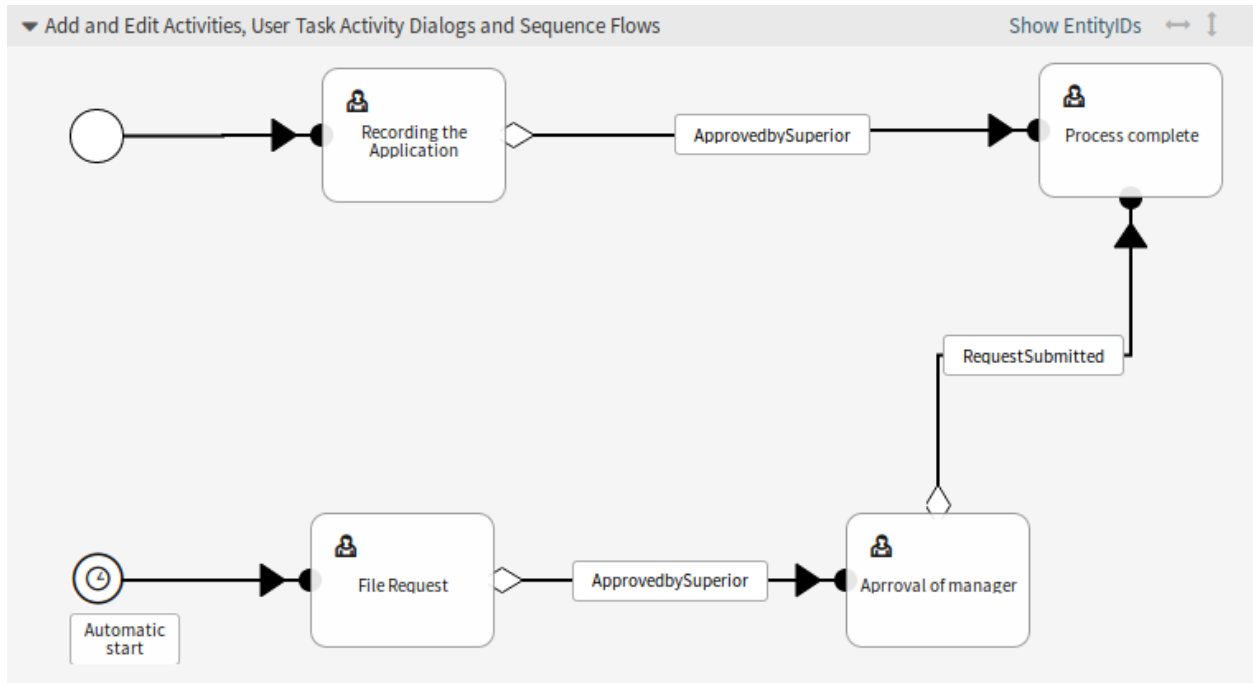


Fig. 4: Start Event on Canvas

7.3 External Interface

This package has no external interface.

Calendar Resource Planning

This feature add-on gives you an operations overview of the entire team. And that's not all. Dynamic fields, reporting functions and notifications can make your work simpler by allowing you to have individualized calendar views, resource planning per person or team, and selected import options into the calendar.

Additionally, customer and ticket data can be synchronized from a linked ticket into an appointment. This data can then be made available to the agent in the field as an offline local calendar. By linking the calendar entries with tickets, you always have complete details about your appointments on hand.

With pre-defined authorization groups, you can even create shared calendars for select teams. Now, important information, such as meetings or deadlines, are available for all relevant agents.

Visualize which agent has which tasks to perform and how much time is estimated for completing these. Meetings, appointments and simple project planning can now be displayed centrally in **OTRS** for the entire department.

Benefits

- Cross-departmental overview of all teams and resources.
- Calendar, resource planning and project planning in one tool.
- Reporting functions and notifications.
- Customer and ticket data also available offline for field service use.

Target Groups

- Customer service
- Facility management
- Human resources
- IT
- IT service management
- Logistics
- Marketing and public relations

- Project management
- Sales

Available in Service Package

- SILVER

Package Name in OTRS Package Manager

- OTRSCalendarResourcePlanning

8.1 Administrator Interface

This package extends the *Calendar Management* screen with a new section to set default values and add dynamic fields to appointments. New management screens for teams and team agents are also available from the *Team Management* and *Team List Management* module of the *Administration* group. It is also possible to send out ICS files in appointment notifications.

8.1.1 Administration

This package extends the *Calendar Management* screen with a new section to set default values and add dynamic fields to appointments. New management screens for teams and team agents are also available from the *Team Management* and *Team List Management* module of the *Administration* group.

Calendars

Use this screen to add default values to appointments. The calendar management screen is available in the *Calendars* module of the *Administration* group.

This package extends the *Calendar Management* screen with a new section to set default title, description and location for appointments.

To set default values for appointments:

1. Click on the *Add Calendar* button in the left sidebar or click on a calendar in the list of calendars.
2. Fill in the required fields.
3. Add default values in the *Manage Settings* widget.

Manage Settings

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

Dynamic Fields

Note: This section is visible only, if *Dynamic Fields* for appointments are added to the system.

Each appointment dynamic fields are listed in this section.

Define for each appointment dynamic field to *Hide*, *Show* or *Show as mandatory*.

▼ Manage Settings

Dynamic Fields

Equipments:

Advanced Settings

Title (default):

Define the default title for appointments using this calendar.
Please note that the defined text is only used during appointment creation (no update functionality), therefore not all OTRS smart tags are usable.

Description (default):

Define the default description for appointments using this calendar.
Please note that the defined text is only used during appointment creation (no update functionality), therefore not all OTRS smart tags are usable.

Location (default):

Define the default location for appointments using this calendar.
Please note that the defined text is only used during appointment creation (no update functionality), therefore not all OTRS smart tags are usable.

Additional description:

Define the additional text which is shown on each appointment in 'Timeline Custom' view to display more information in the overview screens.

It is possible to use OTRS smart tags related to the appointment, e.g. `<OTRS_APPOINTMENT_X>`, `<OTRS_APPOINTMENT_DYNAMICFIELD_X>`.
You can even access data of linked tickets using `<OTRS_APPOINTMENT_TICKET_X>` (if more than one ticket is linked, you can use `<OTRS_APPOINTMENT_TICKET_1_X>`; sorted by ticket ID).

Fig. 1: Manage Settings Widget

Advanced Settings

Title (default) Define the default title for appointments using this calendar.

Description (default) Define the default description for appointments using this calendar.

Location (default) Define the default location for appointments using this calendar.

Additional description Define the additional text which is shown on each appointment in *Timeline Custom* view to display more information in the overview screens. It is possible to show attributes of the appointments, as well as stored dynamic fields. Even the access to linked ticket data is possible.

It is possible to use OTRS smart tags related to the appointment, e.g. `<OTRS_APPOINTMENT_X>`, `<OTRS_APPOINTMENT_DYNAMICFIELD_X>`. You can even access data of linked tickets using `<OTRS_APPOINTMENT_TICKET_X>`. If more than one ticket is linked, you can use `<OTRS_APPOINTMENT_TICKET_1_X>`. Tickets are sorted by ticket ID.

Note: The appointment is not created at this time and not all values are there (e.g. no dynamic field values can be used). The defined texts are only used during appointment creation (no update functionality), therefore not all OTRS smart tags are usable.

The default values are populated whenever the calendar is chosen and the title, description or location is empty.

Behind the input fields for title, description and location, there is a refresh button if default values are specified. This refresh button could be used to re-apply the stored default values if for example dynamic fields are entered in the screen.

Warning: The already entered data will be overwritten using the refresh buttons.

Team Management

Use this screen to manage teams. The team management screen is available in the *Team Management* module of the *Administration* group. Additionally, it is also accessible from the *Team List Management* screen.

Manage Teams

To create a new team:

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

Warning: Teams 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 team:

1. Click on a team in the list of teams.

Manage Teams

Actions

Team Import

Here you can upload a configuration file to import a team to your system. The file needs to be in .yml format as exported by team management module.

No file selected.

Overwrite existing entities

Filter

List

NAME	GROUP	COMMENT	VALIDITY	CHANGED	CREATED	EXPORT
My Team	admin	Custom team for calendars.	valid	07/15/2019 14:23 (Europe/Budapest)	07/15/2019 14:23 (Europe/Budapest)	<input type="button" value="Download"/>

Fig. 2: Manage Teams Screen

Add Team

Team

★ Name:

★ Permission group:

Comment:

★ Validity:

or

Fig. 3: Add Team Screen

2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.

The screenshot shows the 'Edit Team' interface. It features a header 'Team' and a form with the following fields:

- ★ Name: My Team
- ★ Permission group: users
- Comment: Custom team for calendars.
- ★ Validity: valid

At the bottom of the form, there are three buttons: 'Save', 'Save and finish', and 'Cancel'.

Fig. 4: Edit Team Screen

To manage team agents:

1. Click on the *Manage Team Agents* button.
2. Assign agents to teams and vice versa in the *Team List Management* screen.

To export a team:

1. Click on the export icon in the last column of the overview table.
2. Choose a location in your computer to save the `Export_Team_Team_name.yml` file.

The screenshot shows a table titled 'List' with the following data:

NAME	GROUP	COMMENT	VALIDITY	CHANGED	CREATED	EXPORT
My Team	admin	Custom team for calendars.	valid	07/15/2019 14:23 (Europe/Budapest)	07/15/2019 14:23 (Europe/Budapest)	

Fig. 5: Export Team Screen

To import a team:

1. Click on the *Browse...* button of the *Team Import* widget 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 teams.
4. Click on the *Import team* button.

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

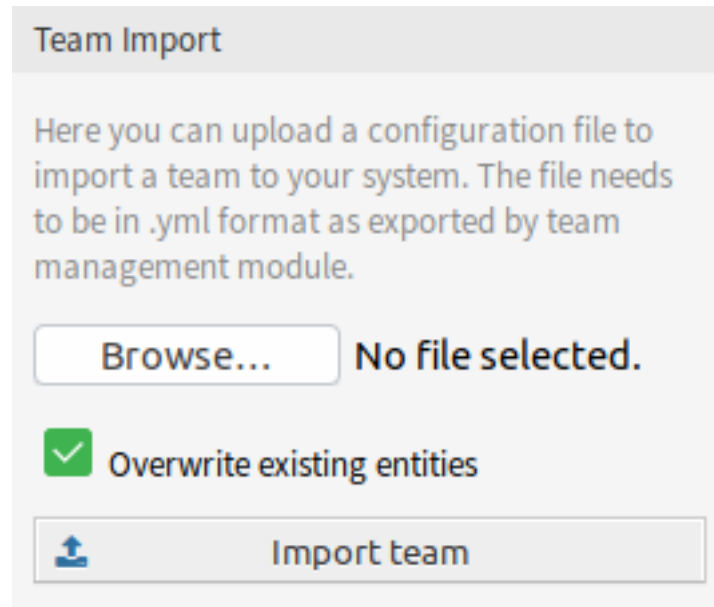


Fig. 6: Team Import Widget

Team 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.

Permission group * Agent groups that can manage the team. Full access to this group is required for anyone to be able to manage the team and its agents, as well as assign them as resources to appointments.

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.

Team List Management

Once you have set up a team in *Team Management* screen, you must define which agents belong to which teams. To use this function, at least one agent and one team need to have been added to the system. The management screen is available in the *Team List Management* module of the *Administration* group. Additionally, it is also accessible from the *Team Management* screen.

Manage Agents Teams Relations

To assign some teams to an agent:

1. Click on an agent in the *Agents* column.

Manage Team Agents

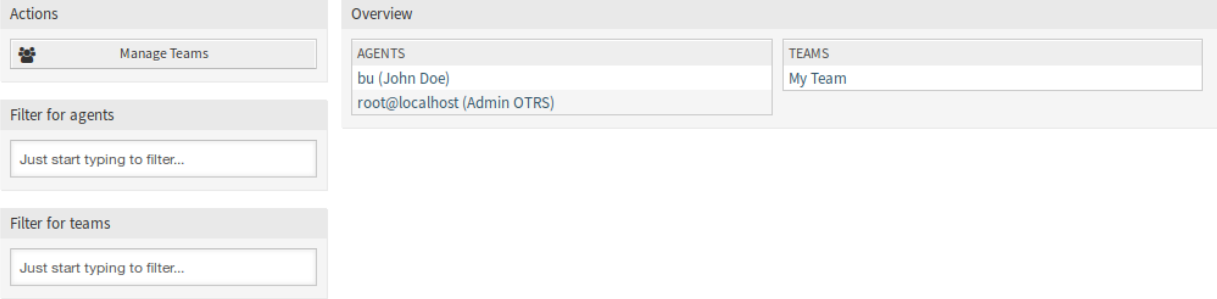


Fig. 7: Manage Team Agents Screen

2. Select the teams you would like to assign the agent to.
3. Click on the *Save* or *Save and finish* button.

Change Team Relations for Agent Doe John (bu)

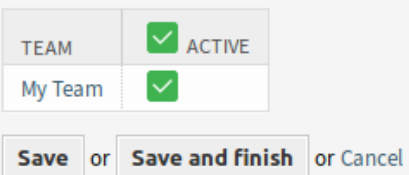


Fig. 8: Change Team Relations for Agent

To assign some agents to a team:

1. Click on a team in the *Teams* column.
2. Select the agents you would like to assign to the team.
3. Click on the *Save* or *Save and finish* button.

Change Agent Relations for Team My Team

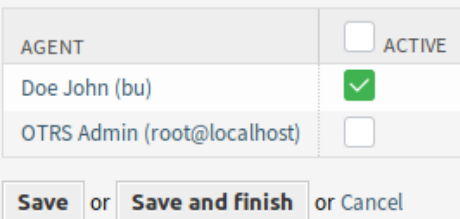


Fig. 9: Change Agent Relations for Team

Note: If several agents or teams are added to the system, use the filter box to find a particular agent or team by just typing the name to filter.

Multiple agents or teams can be assigned in both screens at the same time. Additionally clicking on an agent or clicking on a team in the relations screen will open the *Edit Agent* screen or the *Edit Team* screen accordingly.

Warning: Accessing an agent or a team provides no back link to the relations screen.

8.1.2 Communication & Notifications

This feature adds the possibility to send out ICS files in appointment notifications.

Appointment Notifications

Use this screen to filter appointments and send ICS files in appointment notifications, that can be opened with any regular calendar application. The appointment notification management screen is available in the *Appointment Notifications* module of the *Communication & Notifications* group.

Appointment Filter

Use this section in new or existing appointment notification to filter appointments.

The screenshot shows a form titled "Appointment Filter" with the following fields:

- Calendar:
- Title:
- Location:
- Team:
- Resource:
- Dynamic Fields**
 - Equipments:

Fig. 10: Appointment Filter Section

It is possible to filter based on calendar, title, location, team or resource, as well as dynamic fields (if dynamic field for appointments is created in *Dynamic Fields* screen).

Send ICS Files in Notifications

It is possible to include an ICS file to the appointment notification sent to users.

To activate the sending of ICS files:

1. Create a new appointment notification or select an existing appointment notification.

2. In the *Events* widget select `AppointmentCreate`, `AppointmentUpdate` or `AppointmentDelete` as event.
3. In the *Notification Methods* widget select *Send ICS file*.

You can even configure to use the additional shown information out of the *Custom* timeline view (configurable per calendar) in the ICS file generation as well. Just activate the setting `AppointmentCalendar::ICSFiles::UseAppointmentDescription` to combine the regular description with the configured text.

8.1.3 Processes & Automation

This package adds support for dynamic fields to appointments.

Dynamic Fields

Use this screen to manage dynamic fields for appointments. Dynamic fields for appointments have to be configured in the same way as dynamic fields for other object. Once you have set up the dynamic fields, you have to activate the dynamic fields for calendars.

To activate dynamic fields for calendars:

1. Go to the *Calendars* screen.
2. Select a calendar or create a new one.
3. Find the *Dynamic Fields* section in the *Manage Settings* widget.
4. Define for each appointment dynamic field to *Hide*, *Show* or *Show as mandatory*.

This feature controls only the visibility of the dynamic fields. The dynamic fields have to be added manually to the *Add Appointment* and *Edit Appointment* forms by using the following system configuration settings:

- `Forms###AgentFrontend::Calendar::AppointmentCreate::Properties`
- `Forms###AgentFrontend::Calendar::AppointmentUpdate::Properties`

The dynamic fields can be added to the *Linked Appointments* widget of the detail views as well. The following settings have to be extended:

- `AgentFrontend::TicketDetailView::Widget###LinkedObjects::CalendarAppointment`
- `AgentFrontend::KnowledgeBaseArticleDetailView::Widget###KBALinkedObjects::CalendarAppo`

8.2 Agent Interface

This feature add-on enhances the original calendar implementation by introducing resource allocation for appointments. Furthermore, resources can be assigned to separate teams, which in turn can be used to filter resource overview and availability. The appointments now can be used in list statistics.

8.2.1 Resource Calendar

After installation of the package, a new resource overview widget will be available in the *Calendar Overview* and in the *Dashboard*. While it is possible to assign resources to an appointment in any appointment widget, this widget is designed exclusively for allocating resources to appointments.

This widget looks very much like a regular appointments widget. It has a filter for teams in the widget configuration and a list of agents for that team (available only in time-line views). By choosing a team in the widget configuration, you will be presented with all appointments assigned to the team. A new section *Timeline View “Custom” Settings* is also added to all appointment widgets as a new customizable view.

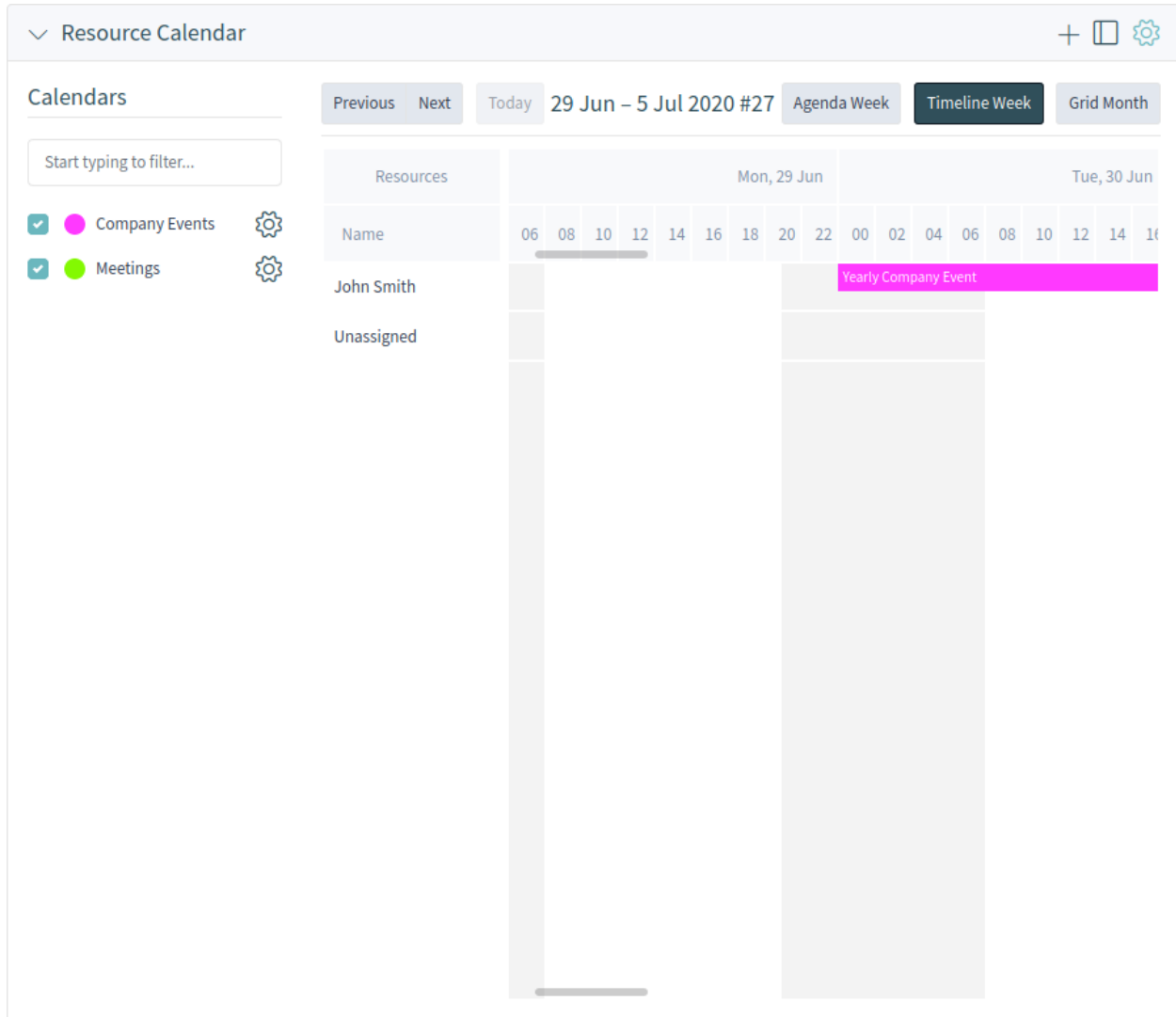


Fig. 11: Resource Calendar Widget

Once you have both teams and their agents set up, you can proceed to a calendar overview and start assigning appointments to teams and agents. Simply add or edit an appointment, and choose both team and agent from the lists under resource. You can do this via any appointment widget, as appointment edit dialog is the same for all of them.

By simply dragging an appointment to a row line with an agent's name in time-line view, it is possible to assign an appointment to this resource. If an appointment does not have an agent assigned to it, it will appear in the *Unassigned* row of the widget. An appointment with multiple assigned agents will be displayed in each row of assigned agents. Dragging one instance of this appointment will move all of them in the same fashion.

By editing the settings of the *Resource Calendar* widget via gear icon in top right corner, you can choose which team agents are displayed within the widget and you can customize the *Timeline View “Custom”*

Settings.

In all other aspects, the *Resource Calendar* widget behaves in the same way as a regular appointment widget.

Timeline View “Custom” Settings

Some of these settings have been applied also to the regular calendar overview.

Edit Widget Configuration
✕

Widget Title

Name your widget

* Grid View

Month
▼

* Timeline View

Week
▼

* Agenda View

Week
▼

* Show Weekends

Yes
▼

* Show Resource Information ℹ

No
▼

✓ Resources

* Teams ℹ

My Team
✕

▼

Resources

John Smith

✓ Timeline View "Custom" Settings

* Shown Weeks

2 Weeks
▼

* Start Time ℹ

00:00
▼

* End Time ℹ

24:00
▼

* Slot Duration ℹ

2 Hours
▼

* Slot Width ℹ

100 Pixel
▼

↻ Reset to Default

Cancel

Save

Fig. 12: Resource Calendar Settings

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

Widget Title Here, the default name of the widget can be changed. If a new name is entered, this name will be displayed in the header of the widget.

Grid View * Here can be selected the time period for the specific view.

Timeline View * Here can be selected the time period for the specific view. The *Resource Calendar* has a new *Custom* view type for timeline view.

Agenda View * Here can be selected the time period for the specific view.

Show weekends * Select whether the weekends should be displayed or not.

Show Resource Information * Show additional resource information next to the agent name in the *Resource Calendar* widget. You can use OTRS smart tags related to agents, e.g. `<OTRS_AGENT_X>`. You can even access agent preferences.

See also:

The additional resource information can be added in setting `AppointmentCalendar::CustomTimeline::Resource` in the system configuration.

Resources * This setting is user and team specific. If you have defined an agent filter in this way, the widget will filter the appointments to the selected teams and agents.

Shown weeks * Select how many weeks should be displayed in the *Custom* timeline view.

Start time * Select the start time for the *Custom* timeline view. The default `00:00` means the start time will be at the very beginning of the day (midnight).

End time * Select the end time for the *Custom* timeline view. The default `24:00` means the end time will be at the very end of the day (midnight).

Slot duration * Select the time slot duration for the *Custom* timeline view. The default `2 Hours` means that each shown time slot has the duration of two hours (e.g. 12 slots are shown for a day 00:00 - 24:00).

Slot width * Select the time slot width for the *Custom* timeline view. The default `100 Pixel` means that each shown time slot has a width of 100 pixel.

In *Custom* timeline view, it is possible to show more information on each shown appointment. For this, you have to configure the setting *Additional Description* in the *Calendars* screen.

Add Appointment

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

Calendar Subscriptions

It is possible to use a filter in calendar subscription feature.

To use the calendar subscription feature:

1. Open the *Calendar Overview* or the *Dashboard* screens.
2. Click on the gear icon next to a calendar in the *Resource Calendar* widget in left sidebar.
3. Select the teams and resources you want to filter for.
4. Click on the *Copy* button. Now the URL is copied to the clipboard.
5. Go to your calendar application and paste the URL.

Resource

Team

My Team x

Agent

John Smith x

Fig. 13: Add Appointment Action

Calendar Subscription: Company Events

Team

Please select team(s) to filter out appointments

Agent

Please select agent(s) to filter out appointments

Cancel Copy

Fig. 14: Calendar Subscription Action

The dialog is kept open after clicking on *Copy* button, to be able to change the selected teams and resources.

8.2.2 Statistics and Reports

After installation of the package you can set up new list statistics using the `AppointmentList` statistic back end to create statistics about the stored appointments in your **OTRS** system. It is a regular statistic (e.g. like the `TicketList` statistic), which means you can configure the X-axis and Y-axis to your needs. A lot of the appointment attributes are even available as restrictions.

Some of the selectable attributes are calculated on-the-fly:

- Age
- Days
- Hours

The filter *Period* could be used to combine start and end time of an appointment with the OR operator instead of AND operator. For example the filter *Start Time* with a relative time period for the current one month will not show appointments starting one month before but ends in the current month. To see such appointments, you have to use the *Period* filter. Appointments which are overlapping are split.

For each linked ticket one row is added to the statistic output. For example an appointment is linked with two tickets, the appointment is shown twice.

Note: Please be careful if you are using teams and resources to filter the appointments. Unfortunately it is not possible to filter for teams and resources on database level, meaning the time which is needed to create a statistic could be higher than usual.

8.3 External Interface

This package has no external interface.

CI Assignment Attribute Dynamic Field Map

This feature add-on makes it possible to link configuration items (CIs) from the OTRS CMDB with services and SLAs, thereby both providing service agents with a better overview and making it possible to offer purely configuration item oriented services. This is especially helpful for companies that provide services for a wide range of different devices or products and that must conform to strict SLAs.

In addition, the feature add-on also makes it possible to map and link configuration item attributes to previously created dynamic fields. If the fields are merely text fields, it is possible to carry out simple mapping in the system configuration. If the configuration item has multiple values that must be displayed in a drop-down dynamic field in the ticket, an extended mapping must be configured.

Benefits

- Better overview for agents thanks to the assignment of configuration items to services and SLAs.
- Enables the provision of configuration item oriented services with numerous configuration item attributes.
- Greater flexibility and transparent linking of configuration item attributes to ticket dynamic fields.
- Optimized visualization of configuration item attributes with multiple values in a drop-down in the ticket.
- Configuration item attributes are readable and searchable.

Target Groups

- Companies that offer services for different devices or products
- Agents
- Internal and external IT
- Facility management

Available in Service Package

- GOLD

Package Name in OTRS Package Manager

- OTRSCIAssignmentAttributeDynamicFieldMap

Note: This feature add-on requires the *Configuration Management* feature.

9.1 Administrator Interface

This package has no administrator interface, but it allows to set dynamic field values as well as service and SLA of tickets based on linked configuration items.

9.1.1 Assign Service and SLA

It is possible to add definition of two configurable fields in configuration items which contain service and SLA information.

Whenever a configuration item is linked to a ticket or unlinked from a ticket, these fields are used to update or remove service and SLA of the linked ticket.

To add the fields for configuration items:

1. Open the *Configuration Items* module of the *CMDB Settings* group in the administrator interface.
2. Select a configuration item class and click on the *Change class definition* button.
3. Add the new fields to the class definition.

```
- Key: TicketServiceName
Name: Service
Searchable: 1
Input:
  Type: Text
  Size: 50
  MaxLength: 50

- Key: TicketSLAName
Name: SLA
Searchable: 1
Input:
  Type: Text
  Size: 50
  MaxLength: 50
```

Note: The `Key` values must be the same as set in `ITSMConfigItem::ServiceField` and `ITSMConfigItem::SLAField` settings.

After this is configured correctly, just create a configuration item with both fields filled out. If you are going to link this newly created configuration item to a ticket, service and SLA from configuration item attributes are taken into the linked ticket.

Requirements and limitations for service and SLA linking:

- Service functionality must be enabled.
- Configuration item fields for service and SLA must be configured.

- Linked configuration item must contain valid service and SLA names.
- Service and SLA names of configuration item must be allowed for the linked tickets (e.g. not restricted via ACL).
- Linking a configuration item when another configuration item is already linked will update service and SLA again.

Requirements and limitations for service and SLA unlinking:

- Service functionality must be enabled.
- Configuration item fields for service and SLA must be configured.
- Service and SLA of ticket and configuration item must match.
- After unlinking, service and SLA of ticket will be removed, not changed to values before linking.

9.1.2 Assign Dynamic Field Values

It is possible to add definition of a configuration item attribute to dynamic field mapping.

Whenever a configuration item is linked to a ticket, it is checked if the configuration item has attributes defined in this mapping and if so, assigns these values to the dynamic fields of the corresponding ticket. Whenever a configuration item is unlinked from a ticket, the dynamic field value remains unchanged.

To configure the mapping:

1. Go to *System Configuration* screen.
2. Select *OTRSCIAssignmentAttributeDynamicFieldMap* in the *Navigation* widget.
3. Navigate to *Core* → *OTRSCIAssignmentAttributeDynamicFieldMap* in the navigation tree.
4. Search for setting `ITSMConfigItem::TicketDynamicFieldMapping` and set the mapping between configuration item attributes and ticket dynamic fields.

For example:

```
NIC::IPAddress          → IPAddress
NIC::IPoverDHCP        → IPoverDHCP
SerialNumber           → SerialNumber
WarrantyExpirationDate → WarrantyExpirationDate
```

Note: If you want to use configuration item attributes nested into a deeper structure, add the chain of attribute keys separated by `::`.

5. Search for setting `ITSMConfigItem::TicketDynamicFieldValueMapping` and set the mapping between configuration item attributes and the values of ticket dynamic fields. We have to specify mappings for this to solve a mismatch between possible values of configuration item attributes and possible values of ticket dynamic fields.

For example:

```
NIC::IPoverDHCP → No   → 2
                  Yes  → 1
```

The numbers define the order in which the values are shown in the drop-down dynamic field. If you would store the values *Yes* and *No* into a dynamic field with the configuration shown above, the system

would try to store the value *Yes* or *No* whereas the dynamic field would expect key 1 for *Yes* or key 2 for *No*.

Note: This mapping can also be used for other fields, if the configuration item attribute value does not match a dynamic field key. In this case write the configuration item attribute value into the key part of the hash and the dynamic field key inside the item tag.

After this is configured correctly, just create a configuration item with the mapped fields filled out. If you are going to link this newly created configuration item to a ticket, the ticket dynamic fields are populated with configuration item attributes. The configuration item attributes will be shown in the ticket information in the right sidebar of the *Ticket Zoom* screen and will be automatically updated when the CMDB is modified.

Limitations of the dynamic field assignment functionality:

- Dynamic fields are just filled if they do not have content yet.
- If a configuration item is unlinked from a ticket, the dynamic field values do not get deleted.
- If a configuration item attribute contains a list of values (for example multiple IP addresses of a computer configuration item) only the first value of this list is assigned to a dynamic field.

9.2 Agent Interface

This package has no agent interface.

9.3 External Interface

This package has no external interface.

With this feature add-on it is possible to enhance the configuration items in the configuration management database (CMDB) through additional fields and to create links and references between these or other data in **OTRS**. This can be useful for linking configuration items with each other when there is a dependency relationship between them or to store information regarding linked services and users directly in the configuration item, as well as find this information with a quicker autocomplete search.

The following input fields with autocomplete search can be added:

- `ReferenceCI` field: a further configuration item.
- `ReferenceService` field: a further service.
- `ReferenceUser` field: a further user.

In the administrator interface, it is possible to create new reference fields under *Configuration Items*. Then, it is necessary to add the visualization configuration as code for the desired configuration item class. If, for example, a new configuration item is being created in the said class, then the newly added reference field will appear in the form to be filled out. A search for the referenced configuration items using autocomplete can then be carried out, and these can be added. If the new configuration item is updated, an automatic linking to the referenced configuration item is carried out.

Benefits

- Enhancement of configuration items through additional input fields and additional, system-referenced data.
- Linking of configuration items with each other and with other data in **OTRS**.
- Quicker autocomplete search for values referenced in configuration item fields.
- Clear visualization of complex dependencies between configuration items and other configuration items, services, and users.

Target Groups

- Configuration item managers
- Internal and external IT

- Facility management
- Sales

Available in Service Package

- GOLD

Package Name in OTRS Package Manager

- OTRSCIReferences

Note: This feature add-on requires the *Configuration Management* feature.

10.1 Administrator Interface

This package has no administrator interface, but it allows to add additional input fields for configuration item classes.

10.1.1 Additional Configuration Item Fields

It is possible to add additional reference fields to refer to different data in **OTRS**, like to other configuration items, services and users.

To add the fields for configuration items:

1. Open the *Configuration Items* module of the *CMDB Settings* group in the administrator interface.
2. Select a configuration item class and click on the *Change class definition* button.
3. Add the new fields to the class definition.

The following sections describe the possible input fields.

ReferenceCI Field

This field adds an input field with auto-completion feature to search for other configuration items in the configuration item dialog. The following example configuration is needed to insert this kind of field:

```
- Key: testci
Name: Test CI
Searchable: 1
Input:
  Type: ReferenceCI
  Required: 0
  Reference:
    Name: Computer
    LinkType: AlternativeTo
    LinkDirection: Source
    ImportExportKey: Name
```

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

Key * Must be unique and only accept alphabetic and numeric characters. If this is changed, data will not be readable from old definitions.

Name *

The label of the field in the form. Any type of characters can be entered to this field including uppercase letters and spaces.

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.

Searchable Defines whether the field is searchable or not. Possible values are *0* or *1*.

Input * Initiates the definition of the input field.

Type * Defines the type of the element. Must be placed indented as a logical block. The value is `ReferenceCI` in this case.

Required Defines whether the field is mandatory or not. Possible values are *0* or *1*.

Reference Initiates the definition of the reference field.

Name Defines the class of the configuration item which should be possible to search for.

LinkType Defines the type of the link which will be created if the value will be saved. Possible values are:

- `DependsOn`
- `AlternativeTo`
- `RelevantTo`
- `Includes`
- `ConnectedTo`

LinkDirection Defines the direction of the link. Possible values are `Source` and `Target`.

ImportExportKey Defines the value for the identification of the referenced configuration item. Possible values are `Name`, `Number` or a configured field key.

After a value is set for the input field the value will be used to set a link to the given configuration item. If there is already a value, then the old value will be unlinked. If the reference field in the class definition has been extended with the setting `CountDefault`, several configuration items can also be linked.

For the export and import of this field the name and the configuration item number are used. If a configuration item is not found for the import, then it will imported 2 times to verify that the linked configuration item which is needed for the link is already imported.

Example export value: `ConfigItemName1`.

ReferenceService Field

This field adds an input field with auto-completion feature to search for services in the configuration item dialog. The following example configuration is needed to insert this kind of field:

```
- Key: testservice
Name: Test Service
Searchable: 1
Input:
  Type: ReferenceService
  Required: 0
  Reference:
    LinkType: AlternativeTo
    LinkDirection: Source
```

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

Key * Must be unique and only accept alphabetic and numeric characters. If this is changed, data will not be readable from old definitions.

Name *

The label of the field in the form. Any type of characters can be entered to this field including uppercase letters and spaces.

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.

Searchable Defines whether the field is searchable or not. Possible values are *0* or *1*.

Input * Initiates the definition of the input field.

Type * Defines the type of the element. Must be placed indented as a logical block. The value is `ReferenceService` in this case.

Required Defines whether the field is mandatory or not. Possible values are *0* or *1*.

Reference Initiates the definition of the reference field.

LinkType Defines the type of the link which will be created if the value will be saved. Possible values are:

- DependsOn
- AlternativeTo
- RelevantTo
- Includes
- ConnectedTo

Additional link types can be defined in the system configuration.

LinkDirection Defines the direction of the link. Possible values are `Source` and `Target`.

After a value is set for the input field the value will be used to set a link to the given configuration item. If there is already a value, then the old value will be unlinked. If the reference field in the class definition has been extended with the setting `CountDefault`, several configuration items can also be linked.

For the export and import of this field the name of the service is used.

Example export value: *Service 1*.

ReferenceUser Field

This field adds an input field with auto-completion feature to search for users in the configuration item dialog. The following example configuration is needed to insert this kind of field:

```
- Key: testuser
  Name: Test User
  Searchable: 1
  Input:
    Type: ReferenceUser
    Required: 0
```

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

Key * Must be unique and only accept alphabetic and numeric characters. If this is changed, data will not be readable from old definitions.

Name *

The label of the field in the form. Any type of characters can be entered to this field including uppercase letters and spaces.

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.

Searchable Defines whether the field is searchable or not. Possible values are *0* or *1*.

Input * Initiates the definition of the input field.

Type * Defines the type of the element. Must be placed indented as a logical block. The value is `ReferenceUser` in this case.

Required Defines whether the field is mandatory or not. Possible values are *0* or *1*.

For the export and import of this field the login of the user is used.

Example export value: *root@localhost*.

10.2 Agent Interface

This package has no agent interface, but the configuration item class definitions might be extended with new fields.

Note: In order to grant users access to the *Asset Management* menu, you need to add them as member to the group *itsm-configitem*.

Use the *Add Configuration Item* menu item in the main menu to add new configuration items to the configuration management database.

To add a configuration item:

1. Select a class from the list of classes.
2. Fill in the required fields.
3. Search for a different configuration item, service or user to set the value for it.
4. Click on the *Save* button.

Now the value is automatically linked to the created configuration item.

See also:

The fields in the *Create Configuration Item* screen can be very different on each classes. The new fields added by this feature add-on is described in the [Administrator Interface](#) chapter.

10.3 External Interface

This package has no external interface.

CIs in Customer Frontend

This feature add-on uses the customer ID attribute of your configuration item classes and makes configuration items visible in the external interface. The customer user has read-only access to the following attributes:

- ID of their configuration item
- Name of the configuration item
- Class of the configuration item
- Deployment state
- Current incident state
- Date and time of the last update

On mail-in this feature add-on automatically links affected configuration items based on the given configuration item ID found in the body of the email. It is also helpful for your service desk as, on ticket creation, they only have access to the configuration items of the requester's organization or department. This eases the selection of affected configuration items a lot, especially if you have many configuration items in your CMDB.

Benefits

- Automatic assignment of configuration items saves time.
- Customers can assign configuration items to new tickets.

Target Groups

- Customer service organizations
- External IT service providers
- Logistics
- Technical field service

Available in Service Package

- GOLD

Package Name in OTRS Package Manager

- OTRSCIsInCustomerFrontend

Note: This feature add-on requires the *OTRS Configuration Management* feature.

11.1 Administrator Interface

This package has no administrator interface.

11.1.1 Extend Class Definition

The definition of a configuration item class has to be altered to make it able to be displayed in the external interface.

By default, the `CustomerID` field is configured in the system configuration to store the customer ID information in the configuration item definition.

You can use more than one `CustomerID` per configuration item, to make one configuration item accessible by more than just one customer. To do so, repeat the following steps for the maximum amount of customers one configuration item may grant access to (for example: `PartnerA`, `PartnerB`, `PartnerC`, etc.).

See also:

To use more than one `CustomerID` or use another name than `CustomerID`, it is necessary to change the setting `ITSMConfigItem::CustomerIDField` in system configuration or add additional entries for each entry field.

If your class definition does not contain the `CustomerID` attribute, then you have to add it manually.

To add the fields for configuration items:

1. Open the *Configuration Items* module of the *CMDB Settings* group in the administrator interface.
2. Select a configuration item class and click on the *Change class definition* button.
3. Add the new fields to the class definition. The input type could be either `Text` or `CustomerCompany`.

Text Field Text based field gives the flexibility to use any string as value to match the `CustomerID` for one or more customers. The value must be entered manually by editing each configuration item, but it needs to be done carefully, because any mismatch will prevent the configuration item to be displayed in the external interface.

```
- Key: CustomerID
Name: Customer Company
Searchable: 1
Input:
  Type: Text
  Size: 50
  MaxLength: 100
```

CustomerCompany Field A customer company field needs to have correctly configured customer companies in the system, as it will be presented as a drop-down list in the configuration item add and edit screens. The source of the drop-down will be the list of companies. `CustomerID` field in all customer users must point to the correct customer ID in customer companies administration.

```

- Key: CustomerID
  Name: Customer Company
  Searchable: 1
  Input:
    Type: CustomerCompany

```

4. Save the new definition.
5. Edit a configuration item from the modified class. Find the *Customer Company* field (or another field you have added) and fill it with the customer ID of an existing customer.
6. Login to the external interface with any customer user that has the customer ID described above.
7. Go to *Company Configuration Items*. The edited configuration item must be listed.

11.1.2 Define Strictness of Customer ID Restriction

The setting `ITSMConfigItem::CustomerCIPermissionByLink` is set to 0 by default, so configuration items are only accessible in the external interface if the company (customer ID) of the customer user matches the value of a configured field. If this behavior is more strict than desired or not all configuration items can/should be configured in such a way, the configuration can be changed to consider links between configuration items (permission inheritance by links). When linking configuration items to new tickets in the external interface and in the agent interface (if enabled), only direct permission is considered though (i. e. `CustomerID` field in configuration item matches).

For example:

- There are multiple computer configuration items assigned to the customer company. All computer configuration items are linked to at least one network configuration item (switches). These devices do not belong to the customer and are therefore not visible. Also the network configuration items are linked to other network configuration items (router).
- By default, only the companies computers are visible under *Company Configuration Items*, are shown in linked tickets and (depending on the configuration) can be viewed in detail and used for new tickets.
- If the setting `ITSMConfigItem::CustomerCIPermissionByLink` is set to 1, all switches connected to a computer will be visible under *Company Configuration Items*, are shown in linked tickets and can be viewed in detail, but not used for to be linked to new tickets.
- If the setting `ITSMConfigItem::CustomerCIPermissionByLink` is set to 2, the routers will be included as well.

Warning: The link type and direction is not relevant for determining the permission. Therefore please carefully consider which value to use for `ITSMConfigItem::CustomerCIPermissionByLink` in order to prevent unwanted disclosure of configuration items.

11.1.3 Hide Configuration Item Fields in External Interface

It could be possible that configuration items has fields that customer does not need to view, or that contains sensitive information that customers must not know. For these cases an administrator can restrict a field by placing a simple new attribute `NotForCustomer` to the field definition on a particular class.

To hide a field in external interface:

1. Open the *Configuration Items* module of the *CMDB Settings* group in the administrator interface.

2. Select a configuration item class and click on the *Change class definition* button.
3. Add the attribute `NotForCustomer: 1` to the field definition.

For example:

```
- Key: Vendor
  Name: Vendor
  Searchable: 1
  Input:
    Type: Text
    Size: 50
    MaxLength: 50
    NotForCustomer: 1
```

4. Edit the configuration items of this class in order to create a new version, so the new version will take the new definition.

To avoid the need to create new versions for configuration items if the definition of a class is updated to hide a field from external interface, it will be applied to all configuration items of that class. On the other hand, if the class definition is updated to show a field that was previously hidden and the configuration item was already updated to the class definition where the field was hidden, the field will not be shown until the configuration item is updated to last class definition where the field is set to be shown again.

The intention of this behavior is to enforce the privacy of the data that should not be displayed in the external interface.

11.2 Agent Interface

This package has no agent interface.

11.2.1 Restriction for Linking

This feature restricts the link interface for the agent when linking a ticket with a configuration item. Only configuration items and the tickets which belong to the same company of the customer user can be selected. The feature can be disabled in the system configuration via the setting `ITSMConfigItem::RestrictAgentLinking`.

11.2.2 Postmaster Filter

There is a postmaster filter added to search incoming emails for a configuration item identifier (usually the configuration item number) and links all found configuration items with the new ticket that was created from this email. The email body and the subject are searched, all found configuration item numbers will be linked with the ticket. This feature is deactivated by default and needs to be enabled in the system configuration via the setting `PostMaster::PostFilterModule###100-ITSMConfigItemLink`.

11.3 External Interface

This feature makes it possible for a customer user to see and select the configuration items of his company in the external interface to link them to new tickets.

11.3.1 Link Configuration Item With Ticket

In the *New Ticket* screen the customer user can search and select the configuration items that are assigned to his company. When the ticket is created, the selected configuration items will be automatically linked to the new ticket.

To link a configuration item to a new ticket:

1. Create a new ticket in the external interface.
2. Fill in the required fields.
3. Click on the *Add affected configuration items* button at the end of the ticket form. A new set of fields will appear below.
4. In the new section fill any search criteria for class, name or number.
 - *Class* drop-down list only shows classes that are already prepared and contain configuration items assigned to a `CustomerID` (for more information take a look at the [Administrator Interface](#)).
 - *Name* field matches any configuration item which name starts with the content of the field. For example, if *ca* is typed in the field, it will match *cat*, *car*, *card*, etc.
 - *ConfigItem#* field matches any configuration item which number starts with the content of the field. For example, if *12* is typed in the field, it will match *123*, *1211*, *1298*, etc.

Note: These search fields will only reduce the search results in order to have smaller lists to be able to select faster. Leaving them in blank will bring the full list of configuration items assigned to the company of the current customer user.

5. Select the configuration items from the list.
6. Click on the *Create Ticket* button.

11.3.2 Company Configuration Items

A new screen *Company Configuration Items* shows a list of all configuration items that belong to the same company as the customer user. Belonging to the same company means, that the configuration items must have a `CustomerID` field that has the same value as the customer user. In the external ticket detail view all linked configuration items that belong to the company of the customer user are shown as well.

If desired, the required association of configuration items to the company of the customer user can be decreased by setting a maximum level to check linked configuration items. If this feature is enabled, configuration items are also accessible to customer users if they do not belong to the company directly, but a linked configuration item is (up to the configured number of links away).

The configuration item details can be access from the configuration items list. This details screen shows the last version of the configuration item, links to other configuration items (if they belong to the same company).

See also:

This screen is not added to any menu of the external interface by default. To create a link to the *Company Configuration Items* screen, an administrator needs to add `/itsmconfigitem/overview` to the *Link* input field and add a name in the *Name* field in one of the following settings of the system configuration:

- `ExternalFrontend::Menu###Top`
- `ExternalFrontend::Menu###Main`
- `ExternalFrontend::Menu###Bottom`

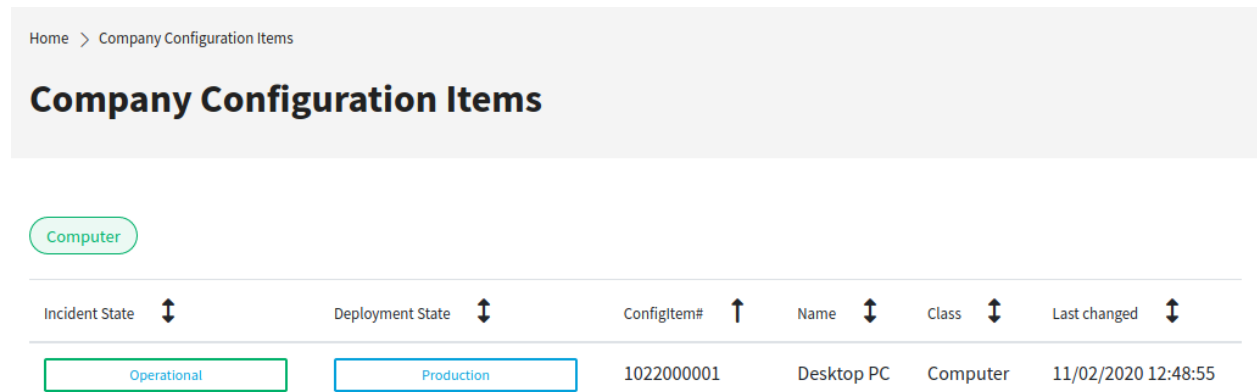


Fig. 1: Company Configuration Items Screen

See also:

The visible columns can be defined in the following setting:

- `ExternalFrontend::ITSMConfigItemOverview###ShowColumns`

Clicking on a configuration item will show the detail view.

Home > Company Configuration Items

Configuration Item: Desktop PC

Configuration Item Details

Customer Company	Super Support Inc.
Owner	"Lacey Green" <lacey@example.org>
CPU	
Ram	
Hard Disk	
↳ Capacity	
Network Adapter	NIC
↳ IP over DHCP	Yes

Configuration Item Information

Class:
Computer

Number:
1022000001

Name:
Desktop PC

Current Deployment State:
Production

Current Incident State:
Operational

Created:
11/02/2020 12:48:55

Created:
11/02/2020 12:48:55

Fig. 2: Configuration Item Details Screen

Configuration Management Connector

This package contains the feature that provides ITSM functionality for the generic interface. It allows to add invokers to request configuration items from remote systems and sync them to the OTRS database.

Additionally this package contains an invoker to sync links between objects from a remote system to the local link database. It is also possible to synchronize configuration items between two OTRS instances using a special synchronization invoker.

Package Name in OTRS Package Manager

- OTRSConfigurationManagementConnector

Note: This feature add-on requires the *Configuration Management* feature.

12.1 Administrator Interface

This package adds a new settings screen to the *Web Service Management* screen to configure the invokers.

12.1.1 Processes & Automation

This package adds three new invokers to *Invokers* section of the *Web Service Management* screen.

Web Services

After installation of the package, three new invokers will be available in the *Invokers* section. Selecting an invoker from the drop-down list will open a new settings window.

Configuration Item Invoker

This package provides the functionality to request a list of configuration items from a remote system with the generic interface. A complete mapping between the name of remote value keys to the local configuration item class definitions is possible. In order to give the possibility to define all class attributes in the mapping, an advanced mapping is provided.

To use this invoker:

1. Go to the *Web Service management* screen.
2. Add a new web service or select an existing one.
3. Select the `ITSM::ConfigItem` invoker in the *Invokers* section.

For handling remote configuration items and links, an advanced mapping is provided with this package. To use this functionality select the `ITSMConfigItem` in the mapping select box.

Invoker Details

★ Name:
 The name is typically used to call up an operation of a remote web service.

Description:

Invoker backend:
 This OTRS invoker backend module will be called to prepare the data to be sent to the remote system, and to process its response data.

Mapping for outgoing request data:
 The data from the invoker of OTRS will be processed by this mapping, to transform it to the kind of data the remote system expects.

Mapping for incoming response data:
 The response data will be processed by this mapping, to transform it to the kind of data the invoker of OTRS expects.

or

Fig. 1: Invoker Details for `ITSM::ConfigItem`

One of those advanced mapping features are static values. These make it possible to define static values for defined keys, for example setting the configuration item class for each item. These entries can overwrite data send from the remote system.

Another feature provided by the advanced mapping are transforming strings to list structures. For creating a list out of a string, a list separator can be defined, for example `;`. The separator field takes a regular expressions which make a more complex separators like `;(?:\s+)?` (a `;` optionally followed by multiple white spaces) possible.

It is possible to define where the index of the elements should take place in the configuration item structure. For this the placeholder `###INDEX###` has to be placed in the key mapping. The configuration in the screenshot would store the separated IP address list into a new interface each. If no index is defined in the mapping, the index will increase the main attribute counter, like a suffix.

The configuration item `Number` attribute is used to make the logical connection between the remote data

Mapping ITSMConfigItem

Default rule for unmapped keys:
This rule will apply for all keys with no mapping rule.

Default rule for unmapped values:
This rule will apply for all values with no mapping rule.

New static value:

★ Key: ★ Value:

New key map:

or or

Fig. 2: Static Mapping for ITSM::ConfigItem

▼ Mapping for Key Interface::###INDEX###::IPAddress

Key mapping: ★ Map key: matching the: ★ to new
key:

List separator:

Value mapping:

New value map:

Fig. 3: Key Mapping for ITSM::ConfigItem

and the data stored in the OTRS database. To create and update configuration items with this invoker values for the following keys are required:

```
- Name
- Class
- Number
- DeploymentState
- IncidentState
```

The values for these keys (except `Number`) can be static or provided by the remote system.

Generic Link Object Invoker

Additionally to the configuration item functionality, this package provides the functionality to link objects, like ticket to configuration items or configuration items to configuration items together. The invoker can get used to pull new links from a remote system or keep the provided links in sync and delete removed links from the OTRS database. There are no limits in linking different objects.

To use this invoker:

1. Go to the *Web Service management* screen.
2. Add a new web service or select an existing one.
3. Select the `Generic::LinkObject` invoker in the *Invokers* section.

For linking different objects together the following mapping keys need values:

```
- SourceClass
- SourceObject
- TargetClass
- TargetObject
- Type
```

The OTRS internal object ID can be provided directly with the key `SourceKey` and `TargetKey` or can get looked up by providing the OTRS object number with the keys `SourceNumber` and `TargetNumber`.

To keep the OTRS in synchronized with the remote system it is possible to select different link combinations, e.g. parent-child ticket to ticket, to get synchronized. This means if a remote link combination was removed the local link will get removed, too.

The synchronization of configuration item classes can be limited by selection the classes that should get synchronized from the list in the invoker administration view.

Configuration Item Synchronization Invoker

This package provides the functionality to synchronize configuration items between two OTRS instances using a special synchronization invoker. That invoker allows to define invokers for create, update and delete synchronizations, that gets called automatically if needed.

To synchronize configuration items between two systems, it is needed to add a web service with an invoker of type `ITSM::ConfigItemSync`. This invoker is the basic (search) invoker that is used to perform ITSM object searches on the remote system to determine configuration items to be modified.

To use this invoker:

1. Go to the *Web Service management* screen.

Mapping ITSMConfigItem

Default rule for unmapped keys:

This rule will apply for all keys with no mapping rule.

Default rule for unmapped values:

This rule will apply for all values with no mapping rule.

New static value:

★ Key: ★ Value:

★ Key: ★ Value:

★ Key: ★ Value:

★ Key: ★ Value:

★ Key: ★ Value:

New key map:

▼ Mapping for Key SourceNumber

Key mapping:
★ Map key: matching the: ★ to new
key:

List separator:

Value mapping:

New value map:

▼ Mapping for Key TargetNumber

Key mapping:
★ Map key: matching the: ★ to new
key:

List separator:

Value mapping:

New value map:

or or

Fig. 4: Mapping Link

Invoker Details

★ Name:
 The name is typically used to call up an operation of a remote web service.

Description:

Invoker backend:
 This OTRS invoker backend module will be called to prepare the data to be sent to the remote system, and to process its response data.

Synchronize links:

Limit CI class sync to:

Mapping for outgoing request data:
 The data from the invoker of OTRS will be processed by this mapping, to transform it to the kind of data the remote system expects.

Fig. 5: Mapping Synchronization

2. Add a new web service or select an existing one.
3. Select the `ITSM::ConfigItemSync` invoker in the *Invokers* section.

It works with two steps on every sync action:

1. At first depending on the change that was made in the local system (add, update or delete a configuration item) the invoker performs a related search on the remote system to determine, if it is needed to create, update or delete a remote configuration item.
2. On the second step, a related invoker for create actions, update actions or delete actions is called, which executes the main changes on the remote system. Those invokers have to be created as well and needs to be of invoker type `ITSM::ConfigItem` to make sure the communication works properly.

See the enclosed overview of example invokers:

Within the search invoker configuration, the worker invokers to perform the changes needs to be configured separately for every action in the related drop-down menus.

If an invoker is not configured for the related action, it will be omitted. The configured invoker will be listed within every drop-down and can be selected easily for the different actions.

The different invokers needs to map their data. For this to work properly, it is recommend to use the mapping module `ITSMConfigItem`, which will be described through this documentation, but in fact it is also possible to use other types of mappings like XSLT if configured properly.

Since this structure calls remote systems, it is needed to configure at least the connection credentials to access the remote CMDB.

All other configurations can be set optional, but they are not needed for the basic feature to work.

To be sure the search invoker just listens on specific events, this package extends the possibilities of the Invoker event filter feature from OTRS, that can be used to add additional conditions to the event triggers, which are configured for the search invoker. One example could be, that just configuration items of ITSM class `Computer` gets synchronized to the remote system.

The configuration of the remote system, that gets synchronized does not need to be special but for it needs to provide at least an operation for the search executions and a separate operation for each sync action that

Invokers

Invokers prepare data for a request to a remote web service, and process its response data.

NAME	DESCRIPTION	CONTROLLER	INBOUND MAPPING	OUTBOUND MAPPING
ConfigItemCreate	Invoker to create remote configuration items (called automatically).	ITSM::ConfigItem	-	-
ConfigItemDelete	Invoker to remove remote configuration items (called automatically).	ITSM::ConfigItem	-	-
ConfigItemSearch	Invoker to search remote configuration items based on local configuration item numbers and perform remote actions.	ITSM::ConfigItemSync	-	-
ConfigItemUpdate	Invoker to update remote configuration items (called automatically).	ITSM::ConfigItem	-	-

Fig. 6: Invoker Overview

needs to be performed.

12.2 Agent Interface

This package has no agent interface.

12.3 External Interface

This package has no external interface.

Invoker Details

★ Name:
 The name is typically used to call up an operation of a remote web service.

Description:

Invoker backend:
 This OTRS invoker backend module will be called to prepare the data to be sent to the remote system, and to process its response data.

Mapping for outgoing request data:
 The data from the invoker of OTRS will be processed by this mapping, to transform it to the kind of data the remote system expects.

Mapping for incoming response data:
 The response data will be processed by this mapping, to transform it to the kind of data the invoker of OTRS expects.

Invoker Create:
 This invoker backend module will be called on create synchronizations

Invoker Update:
 This invoker backend module will be called on update synchronizations

Invoker Delete:
 This invoker backend module will be called on delete synchronizations

Event Triggers:

EVENT	ASYNCHRONOUS	CONDITION	EDIT	DELETE
No data found.				

This invoker will be triggered by the configured events.

Add Event Trigger:

 Asynchronous

To add a new event select the event object and event name and click on the "+" button.
 Asynchronous event triggers are handled by the OTRS Scheduler Daemon in background (recommended).
 Synchronous event triggers would be processed directly during the web request.

or or

Fig. 7: Search Invoker Details

Mapping ITSMConfigItem

Default rule for unmapped keys:
 This rule will apply for all keys with no mapping rule.

Default rule for unmapped values:
 This rule will apply for all values with no mapping rule.

New static value:

★ Key: ★ Value:

★ Key: ★ Value:

New key map:

or or

Fig. 8: Search Invoker Mapping

Operations

Operations are individual system functions which remote systems can request.

NAME	DESCRIPTION	CONTROLLER	INBOUND MAPPING	OUTBOUND MAPPING
ConfigItemCreate	-	ConfigItem::ConfigItemCreate	ITSMConfigItem	ITSMConfigItem
ConfigItemDelete	-	ConfigItem::ConfigItemDelete	ITSMConfigItem	ITSMConfigItem
ConfigItemSearch	-	ConfigItem::ConfigItemSearch	ITSMConfigItem	ITSMConfigItem
ConfigItemUpdate	-	ConfigItem::ConfigItemUpdate	ITSMConfigItem	ITSMConfigItem

Fig. 9: Operation Overview

Custom Contact Fields

This feature add-on makes it possible to create new *customer* dynamic fields, which can store customer data from integrated external customer databases as contacts in a ticket. In an additional configuration interface, it is possible to define whether the new dynamic field type should:

- Store one or more customers.
- Add an additional navigation option in the external interface.
- Include additional contacts in the ticket communication.
- Restrict the adding of new contacts based on customer number or postal codes.
- Be filterable for certain attributes.

Furthermore, event-based notifications can now be configured to be sent to contacts that are stored in the ticket.

This feature add-on is perfect for all businesses that have to deal with different groups of customers, partners or sub-contractors, as it adds, in addition to the customer detail view, a second possibility for storing customer data in a ticket and including it in communication processes.

Benefits

- More flexible and enhanced storage of external customer data in tickets.
- More transparent and automated communication with business partners, sub-contractors or suppliers.
- Seamless integration of external customer databases into **OTRS**.

Target Groups

- Customer service organizations with many partners or sub-contractors
- External IT service providers
- Call centers
- Web, PR or marketing agencies
- Consulting businesses

- Sales departments

Available in Service Package

- SILVER

Package Name in OTRS Package Manager

- OTRSCustomContactFields

13.1 Administrator Interface

This package contains a configuration interface to create dynamic fields of the type *Customer*. These fields can store customer information in a dynamic field and display them in the different screens.

13.1.1 Communication & Notifications

For the external interface, the functionality is by default limited to search just contacts that are from the same company. Otherwise all contacts of all companies would be visible to every customer user that has access to the external interface.

If you want to make all contacts searchable by every customer user, you can enable this by removing the tick from the system configuration option `CustomContactFieldsRestrictContactsToCustomerOnly` and setting it as not enabled.

To activate sending emails to custom contact field customers, go to the *Ticket Notifications* module of the *Communication & Notifications* group in the administrator interface.

Ticket Notifications

Use this screen to activate sending emails to custom contact field customers. The ticket notification management screen is available in the *Ticket Notifications* module of the *Communication & Notifications* group.

Sending Articles Created in the External Interface as Emails

To activate sending emails to custom contact field customer users:

1. Click on the *Add Notification* button in the left sidebar.
2. Fill in the required fields.
3. In the *Add Notification* section, choose a name and set it in the *Name* field.
4. In the *Events* section, select *ArticleCreate*.
5. In the *Article Filter* section:
 - Set agent and customer in the *Article Sender Type* field.
 - Set *Visible to customer* in the *Customer visibility* field.
 - Select *Email* in the *Communication channel* field.
 - Switch *Include attachments to notification* to *Yes*.
6. In the *Recipients* section set a dynamic contact field in the *Send to* field.
7. In the *Notification Methods* section:

- Make sure, that the *Email* method is enabled.

Note: Joined notifications are only sent by *Email*.

- Select the checkbox in the *Article visible to customer* field.

8. In the *Notification Text* section:

- If you want the contacts of the custom contact field to get the title of the article as mail subject add `<OTRS_CUSTOMER_SUBJECT>` into the *Subject* field.
- If you want to send the article body as mail body add `<OTRS_CUSTOMER_BODY>` into the *Text* field.

9. Click on the *Save* button.

13.1.2 Processes & Automation

This package makes it possible to create dynamic fields of the type *Customer*, which can store customer users as contacts in tickets.

Dynamic Fields

After installation of the package a new dynamic field type *Customer* will be available for the objects.

This dynamic field can be created the same way as default dynamic fields are created. For this navigate to the *Dynamic Fields* module of the *Processes & Automation* group in the administrator interface. In this screen you can select the *Customer* field from the drop-down lists on the left side.

See also:

The usage of dynamic fields and the general dynamic field settings are described in the [administrator manual](#).

Dynamic Field Settings

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

Customer Dynamic Field Settings

Customer dynamic field is used to store a customer user.

Input type * Defines whether the dynamic field should be able to store a *Single contact* or *Multiple contacts*.

Navigation on external interface * Defines whether or not a new navigation entry in the external interface should be shown. This navigation entry will only be shown to customer users that are stored in one of the *Customer* fields. The label of this entry can be defined in the setting *Navigation label* if this setting is saved with *yes*.

Navigation label This configuration makes it possible to define a custom label for the navigation entry displayed in the external interface. If no value is configured, a default combination of the field label and *Tickets* will be used, for example *Contacts Tickets*.

Customer Field Settings

<p>★ Input type: <input type="text" value="Single contact"/></p> <p>Choose if the input should support a single contact or multiple contacts per field.</p>	<p>★ Use for <input type="text" value="No"/></p> <p>communication: Change this to a communication field, if you want to use this contact for communications.</p>
<p>★ Navigation on <input type="text" value="No"/></p> <p>external interface: Change this, if you want to display a tab in the external interface where the customer can see all the tickets where he is stored in this dynamic field.</p>	<p>★ Use for <input type="text"/></p> <p>notification: Change this to a communication field, if you want to use this contact for notifications.</p>
<p>Navigation label: <input type="text"/></p> <p>Defines the name of the tab in the external interface. If no name is set the dynamic field name will be used. For example: Contacts tickets.</p>	<p>★ Filter contact by: <input type="text" value="No"/></p> <p>This configuration defines if the contacts should be filtered by one of the available customer attributes.</p>

Fig. 1: Customer Dynamic Field Settings

Use for communication * If customer users stored in a dynamic field of type *Customer* should get used for communication, this configuration has to be used. It is possible to use customer users for the *To*, *Cc* and *Bcc* field. If this configuration is active, the customer user will be added to the configured field. If the field is already containing the address of the customer user, it will not get added a second time. The agent still has the opportunity to remove the address from the field when composing a new message.

Use for notification * If customer users stored in a dynamic field of type *Customer* should get used for notification, this configuration has to be used. It is possible to use customer users for the *To*, *Cc* and *Bcc* recipient. If this configuration is active, the dynamic field will be shown in the recipient block of the ticket notifications.

Note: If the configuration option `UserForNotification` is not set in the dynamic field table, the fallback recipient will be used. The fallback recipient is the *To* recipient.

Filter contact by * In some scenarios it is necessary to filter possible contacts by their attributes. A filter for example could be the customer number, the city or a custom attribute. To activate the filter functionality, select the needed attribute from the drop-down list. If the customer number is selected, the contact will be filtered by the customer number of the customer of the ticket. For each other attribute the filter can be defined as a text value. After selecting the filter attribute from the drop-down list, a second input field will be displayed where the filter can be defined.

Filter criteria * In this field the filter of the attributes for the customer result is defined. For example, if only customer users with the first name *Theo* should be selectable, the filter *Fistname* from the *Filter contact by* has to be selected. After that the *Filter criteria* field will be displayed. Now we can insert *Theo* into the filter field. Currently it is not possible to define a regular expression or a placeholder for filtering customer users.

Note: Do not forget to add the new dynamic field to the desired screens.

13.2 Agent Interface

This package has no agent interface.

13.2.1 New Features

Storing/changing customer user in a *Customer* field After the dynamic field of the type *Customer* was configured in the *Dynamic Fields* screen and added to the different views, the functionality can get used. The contact fields will be displayed below the recipients for new tickets or otherwise in the dynamic fields block of the page. The exact position and label of the field depends on the field configuration.

If a customer number filter was configured the dynamic field will be locked and unchangeable as long as no customer is added to the ticket.

After inserting at least one character into the text field of the dynamic field a search over the configured customer databases starts. At this point the configured filter takes it place and removes unmatched customer users from the result set. To start a search for all available customer users for this field you can use the * wildcard.

Showing *Customer* dynamic field values for a ticket To allow display of the *Customer* dynamic field values after the data was stored to the ticket, the different views have to be configured accordingly.

Using *Customer* dynamic field values for communication in a ticket If a field has been configured to use the contacts for communication and the *Reply to All via Email* action is used in the ticket detail view, all customer users stored in the respective field will be added to the configured recipient list (to, cc or bcc) unless they exist as a recipient already. If necessary, they can be removed from the recipients manually. This functionality is not available for the reply to action.

Note: The field must not be configured in the dynamic field section of the reply actions.

Searching tickets for *Customer* dynamic field values To allow searching of the *Customer* dynamic field values after the data was stored to the ticket the search view has to be configured accordingly.

The search works equals to the search for regular customer users. The login name of the customer user has to be used as search criteria.

13.3 External Interface

This package has no external interface.

To display the dynamic field of type *Customer* in the external interface, see the *Dynamic Fields* chapter.

Customer Frontend Link Object

With this feature add-on, linked knowledge base articles, tickets and other objects, can be displayed in the ticket detail view of the external interface. In the system configuration, you can easily select whether you want to display linked tickets and/or knowledge base articles in the external interface.

In addition, the feature add-on ensures that only those linked tickets for which the customer user has read or write access rights are displayed. If this is not the case, linked tickets are not displayed to stop the customer user from receiving unnecessary error messages.

Benefits

- More transparent communication with customers.
- More efficient workflow to display knowledge base articles in the external interface.
- Gives customers greater insight into the incident-solving process.

Target Groups

- Internal and external IT
- Customer service
- Facility management

Available in Service Package

- SILVER

Package Name in OTRS Package Manager

- OTRSCustomerFrontendLinkObject

14.1 Administrator Interface

This package has no administrator interface.

14.2 Agent Interface

This package has no agent interface.

To use this functionality on *External Interface*:

1. Create two tickets on the agent interface.
2. Choose the same customer user for both tickets.
3. Link one ticket to the other one.
4. Create a knowledge base article. Make sure that the knowledge base article is visible to customers or public. Just knowledge base articles of this type can be displayed in the external interface.
5. Link the knowledge base article to the ticket created for the customer user.

14.3 External Interface

This package has no external interface, but it shows linked tickets and knowledge base articles in ticket detail view.

To use this functionality, an agent has to create tickets and knowledge base articles in the *Agent Interface* and link them together.

To see the linked object in external interface:

1. Log in as customer user.
2. Select *My Tickets* from the ticket menu.
3. Select a ticket that has linked ticket and/or knowledge base article.
4. See the widget containing the linked objects in the right sidebar of the ticket detail view.

Clicking on a ticket or knowledge base article will lead you to the linked ticket or knowledge base article.

Customer-specific Services

This feature add-on makes it possible to assign services to customers so that only the assigned service is displayed when a ticket is created and only the appropriate SLAs are valid.

After installation of the package, a new module *Customers Service* appears in the *Users, Groups & Roles* group in the administrator interface. Here you can define which services should be assigned to customers. Furthermore, you can define in the system configuration which of these options should be the leading one.

Benefits

- Efficient and well-structured administration of multiple services and SLAs for many customers.
- Avoid the creation of unwarranted tickets that stress out your service team.
- Optimal guidance for the customer regarding for which services he or she can open a ticket.
- Speed-up of ticket handling process.

Target Groups

- Internal IT departments
- External IT service providers
- Call centers
- Agencies
- Consulting businesses
- Companies offering a wide range of services or products

Available in Service Package

- SILVER

Package Name in OTRS Package Manager

- OTRSCustomerSpecificServices

15.1 Administrator Interface

After installation of the package a new module will be available in the *Users, Groups & Roles* group of the administrator interface.

Note: The determination of the customers is based on a customer company search. So it is very important to adjust the `CustomerCompanySearchListLimit` setting in the `CustomerCompany` section in the `Kernel/Config.pm`. Otherwise not all customers can be found and displayed.

15.1.1 Users, Groups & Roles

After installation of the package, a new module *Customers Service* appears in the *Users, Groups & Roles* group in the administrator interface. Here you can define which services should be assigned to customers.

Customers Services

Use this screen to add one or more customers to one or more services. To use this function, at least one customer and one service need to have been added to the system. The management screen is available in the *Customers Services* module of the *Users, Groups & Roles* group.

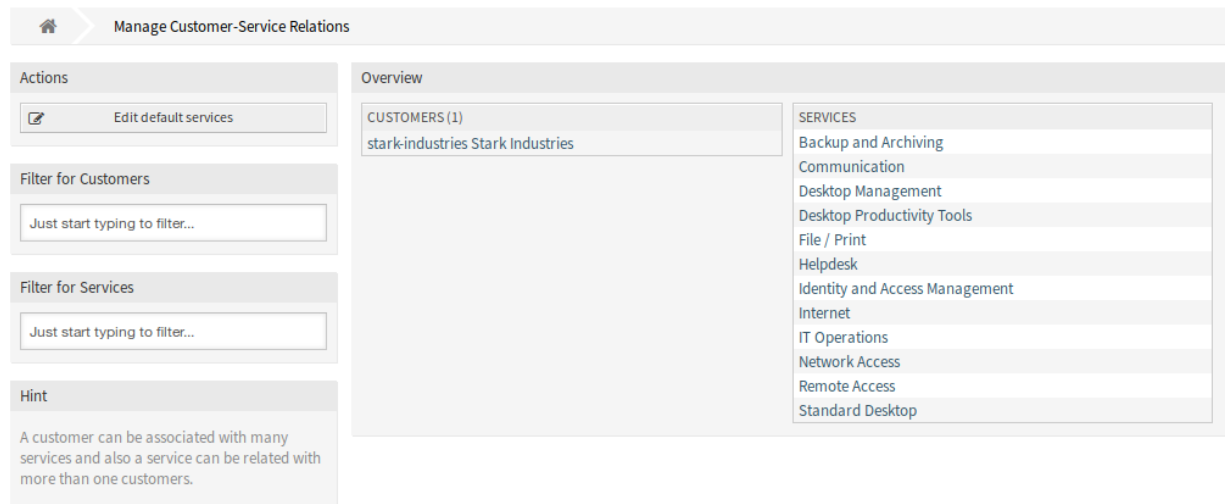


Fig. 1: Manage Customer-Service Relations

Manage Customers Services Relations

Note: Before your first use, please make sure that you activate the *Services* option in the administrator interface.

To allocate some services to a customer:

1. Click on a customer in the *Customers* column.

2. Select the services you would like to allocate to the customer.
3. Click on the *Save* or *Save and finish* button.

Allocate Services to Customer stark-industries

SERVICE	<input type="checkbox"/> ACTIVE
Backup and Archiving	<input checked="" type="checkbox"/>
Communication	<input type="checkbox"/>
Desktop Management	<input type="checkbox"/>
Desktop Productivity Tools	<input type="checkbox"/>
File / Print	<input checked="" type="checkbox"/>
Helpdesk	<input type="checkbox"/>
Identity and Access Management	<input type="checkbox"/>
Internet	<input checked="" type="checkbox"/>
IT Operations	<input type="checkbox"/>
Network Access	<input type="checkbox"/>
Remote Access	<input type="checkbox"/>
Standard Desktop	<input type="checkbox"/>

Save or **Save and finish** or Cancel

Fig. 2: Change Service Relations for Customer

To allocate some customer to a service:

1. Click on a service in the *Services* column.
2. Select the customers you would like to allocate to the service.
3. Click on the *Save* or *Save and finish* button.

Allocate Customers to Service Backup and Archiving

CUSTOMER ID (1)	<input checked="" type="checkbox"/> ACTIVE
stark-industries Stark Industries	<input checked="" type="checkbox"/>

Save or **Save and finish** or Cancel

Fig. 3: Change Customer Relations for Service

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

Multiple customers or services can be assigned in both screens at the same time. Additionally clicking on a customer or clicking on a service in the relations screen will open the *Edit Customer* screen or the *Edit Service* screen accordingly.

Warning: Accessing a customer 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.

Note: The allocations of customer users and services should be done with this screen. It is not recommended to use the screen *Customer Users Services*.

Manage Default Services

It is possible to add default services, so that all customer users of the customer may access them. This prevents having to add each service to each customer.

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.
3. Click on the *Save* or *Save and finish* button.

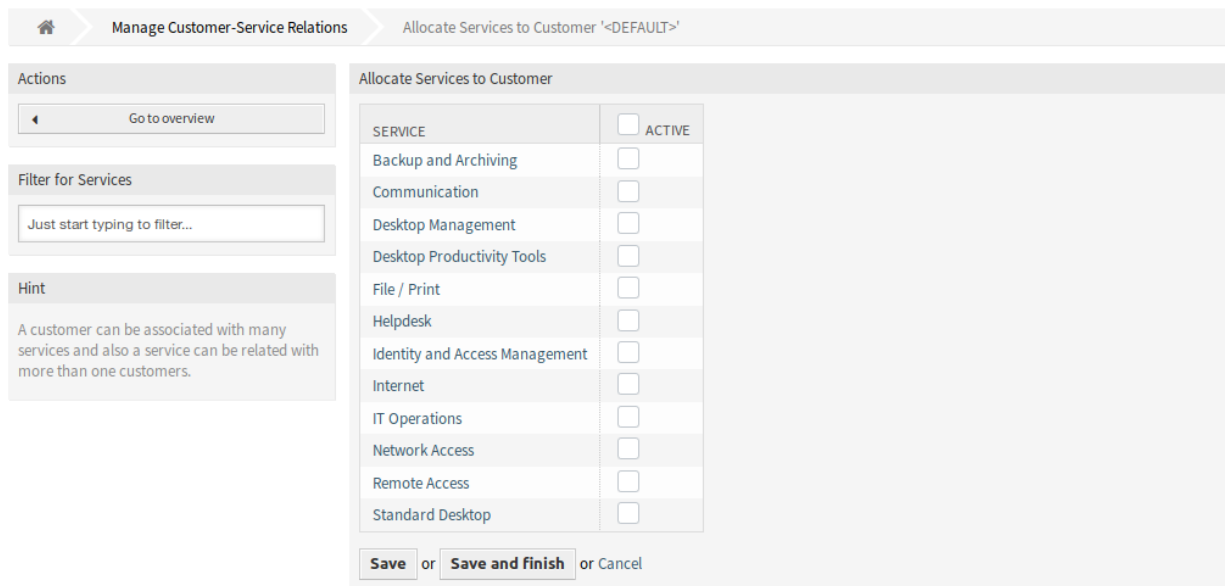


Fig. 4: Allocate Services to Customer Screen

Warning: Mixing default services and customer specific services can be confusing. If a customer has specific services assigned, then the default services will be not applied.

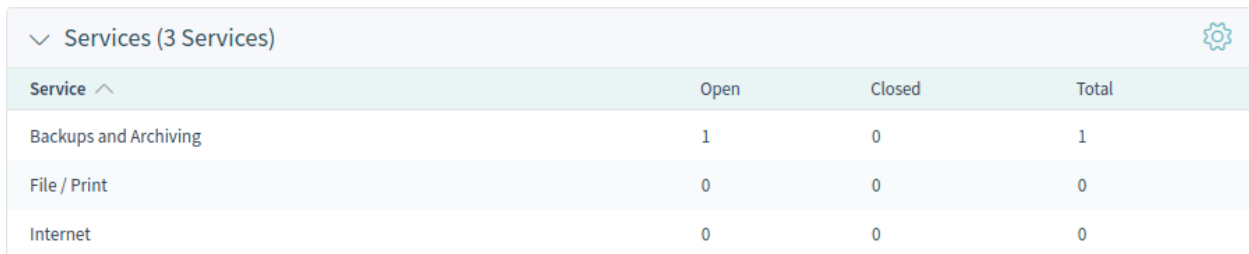
15.2 Agent Interface

After installation of the package a new widget will be available in the customer detail view.

When selecting a customer user in any new ticket screens, only the services related to the customer of that customer user will be available.

15.2.1 Customers

After installation of the package, a new widget *Services* appears in the customer detail view. In this widget all assigned services and the amount of open and closed tickets are displayed.




Services (3 Services) 			
Service ^	Open	Closed	Total
Backups and Archiving	1	0	1
File / Print	0	0	0
Internet	0	0	0

Fig. 5: Services Widget

15.3 External Interface

This package has no external interface, but the service selection can be restricted to the configured ones.

15.3.1 Restrict Service Selection

See also:

The `ExternalFrontend::TicketCreate###Service` setting needs to be enabled to use this feature.

In the *New Ticket* screen only the services which are related to the customer of the customer user will be available. After the service is chosen by the customer user, the associated service level agreements (SLA) will become available, if they are configured.

Dashboard News Widget

Internal news or information can be added via the administrator interface to appear with the corresponding *Announcements* widget in the agent interface. A click opens the detailed view of the respective news item, containing the title, content, creation date and change date. The widget can be individually activated or deactivated as an optional widget in the agent dashboard. The *Announcements* widget makes it easier to inform employees about events, innovations and offers.

Benefits

- Integrates your company communication in the customer service tool.
- Important information, such as product news, is always available to agents and can be used immediately in customer contact.
- Company news is clearly structured and centrally available in one place.

Target Groups

- IT security
- Internal and external communication
- Human resources
- Internal and external IT

Available in Service Package

- SILVER

Package Name in OTRS Package Manager

- OTRSDashboardNewsWidget

16.1 Administrator Interface

This chapter describes the new features that are available in the administrator interface after installation of the package.

16.1.1 Communication & Notifications

After installation of the package a new module will be available to *Communication & Notifications* group in the administrator interface.

Dashboard News

Use this screen to add announcements to agent dashboard. The announcement management screen is available in the *Dashboard News* module of the *Communication & Notifications* group.

Dashboard News Management

TITLE	ANNOUNCEMENT	CHANGED	CREATED	DELETE
Version 1.0 is released	We are pleased to announce the first public ver...	04/04/2019 12:45 (Europe/Budapest)	04/04/2019 12:45 (Europe/Budapest)	
Documentation	The documentation is finished soon.	04/04/2019 12:43 (Europe/Budapest)	04/04/2019 12:43 (Europe/Budapest)	

Fig. 1: Dashboard News Management Screen

Manage Announcements

To add an announcement:

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

Add announcement

★ Title:

★ Announcement:

Save or Cancel

Fig. 2: Add Announcement Screen

To edit an announcement:

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

Edit announcement

★ Title:

★ Announcement:

or or

Fig. 3: Edit Announcement Screen

To delete an announcement:

1. Click on the trash icon in the list of announcements.
2. Click on the *OK* button in the confirmation dialog.

List ⚙️ 1-2 of 2

TITLE	ANNOUNCEMENT	CHANGED	CREATED	DELETE
Version 1.0 is released	We are pleased to announce the first public ver...	04/04/2019 12:45 (Europe/Budapest)	04/04/2019 12:45 (Europe/Budapest)	
Documentation	The documentation is finished soon.	04/04/2019 12:43 (Europe/Budapest)	04/04/2019 12:43 (Europe/Budapest)	

Fig. 4: Delete Announcement Screen

To limit the number of displayed announcements per page:

1. Click on the gear icon in the top right corner of the overview header.
2. Select the maximum number of announcements displayed per page.
3. Click on the *Submit* button.

Announcement Settings

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

Title * The title of the announcement. This will be displayed in the dashboard widget of the agent interface.

Announcement * The body text of the announcement. This will be displayed in the overview screens and in the announcement details screen.

Rich Text editor can be enabled for the announcement text.

See also:

The `DashboardNewsWidget::RichTextField` setting needs to be enabled to use this feature.

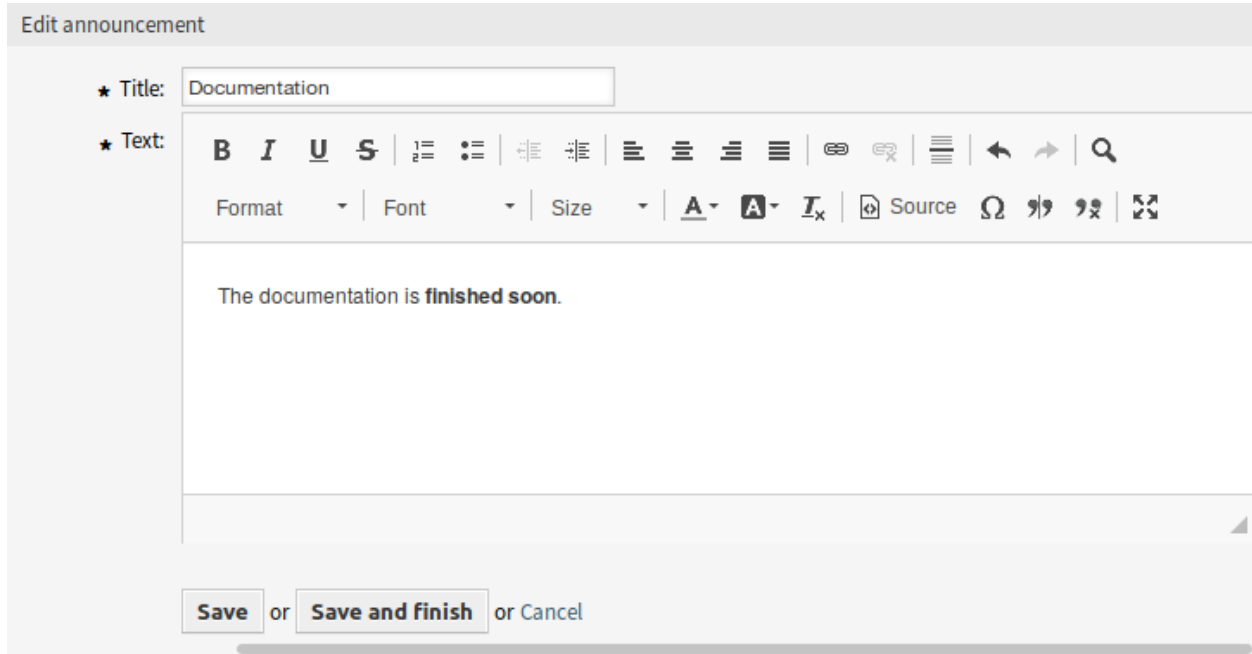


Fig. 5: Edit Announcement Screen with Rich Text Editor

Warning: If announcements were created using a Rich Text editor and the way to manage announcements was changed to normal text in the meantime, the content of the related announcements will be displayed with HTML tags and should be improved afterwards.

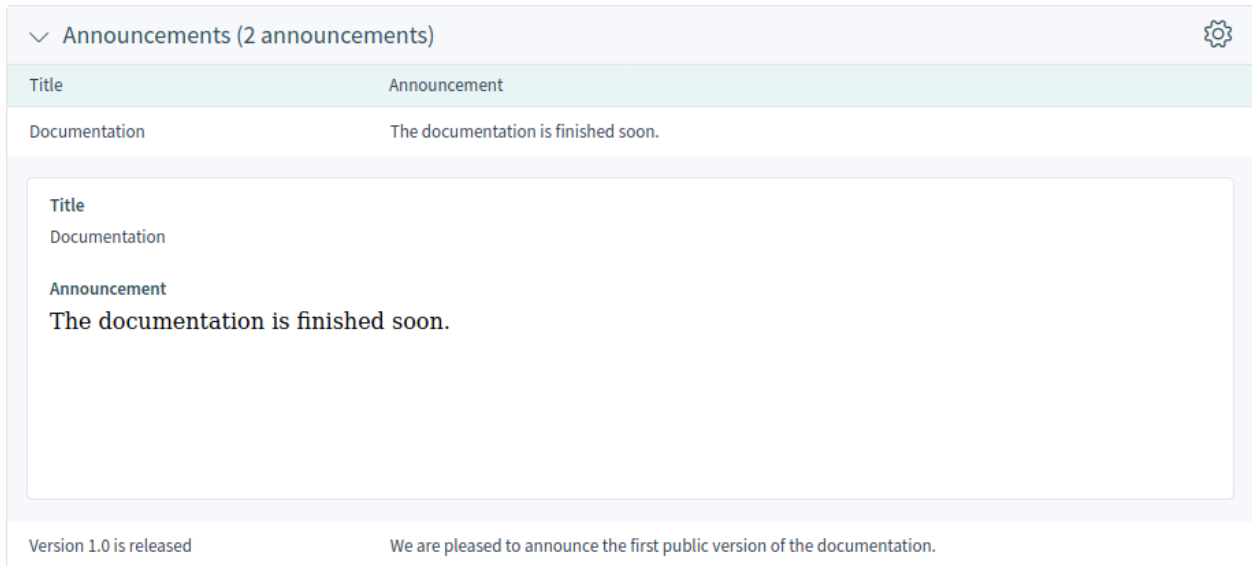
16.2 Agent Interface

After installation of the package a new widget will be displayed in the agent dashboard. Additionally, a new menu group will be displayed in the new action menu to add announcements. This menu item opens the *Add Announcement* screen of the *Dashboard News* administration module.

Announcements (2 announcements) ⚙️	
Title	Announcement
Documentation	The documentation is finished soon.
Version 1.0 is released	We are pleased to announce the first public version of the documentation.

Fig. 6: Announcements Widget

Clicking on an entry in the widget will expand the announcement to see the details.



The screenshot displays the 'Announcements (2 announcements)' section in the OTRS interface. At the top, there is a header with a dropdown arrow and a gear icon. Below this, a table lists the announcements. The first row shows 'Title' as 'Announcement' and 'Documentation' as 'The documentation is finished soon.'. A large, light-colored box below the table contains a preview of the announcement content, showing the title and documentation text. At the bottom of the interface, there is a footer with two columns of text: 'Version 1.0 is released' and 'We are pleased to announce the first public version of the documentation.'

Title	Announcement
Documentation	The documentation is finished soon.

Title
Documentation

Announcement
The documentation is finished soon.

Version 1.0 is released We are pleased to announce the first public version of the documentation.

Fig. 7: Announcement Details

16.3 External Interface

This package has no external interface.

Data Privacy Protection

This feature add-on takes care about the data privacy of specific information within **OTRS**. Based on customizable rules, different actions can be performed on different types of data.

Warning: The use of this feature add-on can result in a complete loss of data! We recommend a backup of the database **before** execution and if possible a test on a test system.

Warning: The use of this feature add-on is at your own risk. *OTRS AG* does not take any responsibility for lost data.

Benefits

- Compliance with data protection guidelines.

Target Groups

- IT service management
- Human resources
- Internal IT
- Government
- IT service
- Security management

Available in Service Package

- TITANIUM

Package Name in OTRS Package Manager

- OTRSDataPrivacyProtection

17.1 Administrator Interface

This package has no administrator interface. All data privacy protection rules must be added via the system configuration and executed by a console command.

17.1.1 System Configuration Settings

For security reasons, this package does not come with any pre-configured rules. Therefore the functionality does not work out-of-the-box and the rules must be configured by an administrator first.

To add a rule configuration:

1. Go to the *System Configuration* screen of the administrator interface.
2. Select *OTRSDataPrivacyProtection* in the *Navigation* widget.
3. Navigate to *Core* → *OTRSDataPrivacyProtection* in the navigation tree.
4. Add a rule configuration in YAML format to setting `OTRSDataPrivacyProtection::RuleConfiguration`.

The configuration of the individual rule sets are stored in YAML format and consists of the five options. The options marked with an asterisk are mandatory.

RuleName * This string option has to be unique for all rules. It is a string, that is used to identify every different rule. The rule name appears in all related outputs and history entries, but does not affect any functionality.

RuleSource This string option is just for informational usage. Different countries and regions have different types and kinds of data protection regulations, that might be part of laws and/or written papers. To identify the related original sources, the different configured rules are based on. `RuleSource` can be used to add names and/or descriptions, that will be added to related history entries (if available).

RuleType * This string option describes the action, that will be executed on all objects, that were found during a search. The following actions are supported (in short format or in long format):

Anonymization Or PrivacyByAnonymization The anonymization rule type is used to anonymize data sets, as identified in the data classification fields. Anonymization means, that the different fields will be replaced by a string *Anonymized*.

Warning: During anonymizations, the original data sets are replaced by the mentioned string in the database. The original data is not saved and is therefore irretrievably lost!

Pseudonymization Or PrivacyByPseudonymization Like the anonymization, the pseudonymization rule type is used to remove the related data from their original fields. There is a major difference to the `Anonymization` rule type, as the data will be stored in a separated database table, called `data_pseudonymization`.

This table is not used by any other sub-system, and it is not usable via the GUI. It acts as a backup table, that can be (manually) searched by any administrator, that has access to the database or using the *SQL Box* module of the administrator interface.

During pseudonymizations, a universal unique identifier (UUID) will be created for the related data field, which is used to identify the original data later. The original data will then be copied to the backup table, using the UUID as the field identifier. Afterwards the original data will be replaced by just the UUID, which is effectively the same as an anonymization, but including a pointer to the stored original source data.

Deletion Or PrivacyByDeletion The deletion rule type is used to delete data sets, as identified in the data classification fields. Deletion means, that the different fields will be replaced by a string *Deleted*. Technically the original data will be deleted, as it will be replaced with a non-sensitive string, it therefore works equal to the *Anonymization* rule type.

Warning: During deletions, the original data sets are replaced by the mentioned string in the database. The original data is not saved and is therefore irretrievably lost!

DataClassification * This list option is used to identify the data types for which the associated actions are to be applied. It contains the different fields of any data object as an array. Every object drivers provides a list of possible data classification fields, that might be used.

See also:

The specific fields are described in the *Drivers* section below.

ObjectFilter * This list option implements the search and filter criteria for every used driver. Every driver provides a list of possible search and filter options, that are allowed to be used.

See also:

Please see the *Drivers* section below for more detailed information.

The different types of information represented by objects (e.g. ticket, customer user, etc.) are called *object types* within this package. Therefore we are talking about *object types*.

The modules implemented for the specific data processing, search functions and verification of such specific object types are called *drivers* or *driver objects*.

17.1.2 Rule Execution

Once the rules have been defined, they can be applied to the existing data sets. There is a console command `Maint::DataPrivacy::Execute` for this.

Execute the console command with the `--help` option for more information:

```
otrs> /opt/otrs/bin/otrs.Console.pl Maint::DataPrivacy::Execute --help
```

This command essentially offers three different options:

- Checking the integrity and validity of the existing rules.
- Test execution of existing rules without changing data records.
- Execution of the existing rules, whereby the matching data sets are permanently changed.

The validation checks all available rules in the context of the affected drivers and object types. If certain options are missing or incorrect, the rule is declared invalid and execution is skipped for all drivers.

For security reasons, the validity of the corresponding rules is implicitly checked both before each dry run and before each execution and is either completely stopped or skipped in case of errors.

Warning: We recommend, that new rules or significant changes should be executed on test systems first to ensure that no data is accidentally changed or deleted.

Warning: We recommend that you back up the database first to ensure that untested data is not lost after rules or rule changes are executed.

Warning: Since rules are designed to change or completely delete data, it is very important to carefully check all rules in advance and execute the test runs for each rule change.

The console output of rule executions can be redirected into files to preserve the modified objects. Please see the following example:

```
otrs> /opt/otrs/bin/otrs.Console.pl Maint::DataPrivacy::Execute --execute-  
↪detail > rule-execution.txt
```

17.1.3 Drivers

This section describe the configuration and usage of the various drivers. In addition, this section contains sample configurations that can be copied and customized to your personal needs.

Customer Company Driver

The customer company driver provides the functionality to search and modify the information for customer companies.

Possible data classifications:

```
- CustomerID  
- CustomerCompanyName  
- CustomerCompanyCountry  
- CustomerCompanyStreet  
- CustomerCompanyZIP  
- CustomerCompanyCity  
- CustomerCompanyURL  
- CustomerCompanyComment  
- DynamicField_NameX
```

The driver supports dynamic fields for data classification. Dynamic fields will be identified by the prefix `DynamicField_` and the related field name.

Possible object filters:

```
- ValidID  
- CustomerID  
- CustomerCompanyStreet  
- CustomerCompanyURL  
- CustomerCompanyComment  
- WildcardSearch  
- CustomerCompanyZIP  
- CustomerCompanyCountry  
- CustomerCompanyName  
- CustomerCompanyCity
```

Object filter descriptions:

- **Limit:** Limits the number of search results.
- **CreateTime:** Searches for dates *greater than or equal to* (>=) the given time.
- **WildcardSearch:** Affects all object filters, except `ValidID`. It will wrap the search term in wildcards, so they match all objects that contain the supplied value. For example, if this option is set to 1, filtering on `CustomerID` with `company` will in fact search for `*company*`.

Rule Configuration Examples

Here are some examples for rule configurations. These examples are valid YAML codes. You can copy these examples and modify them according to your needs.

Delete customer company name and customer company country by customer company name without wildcard search:

```
---
RuleName: Delete customer company name and customer company country by
↳customer company name without wildcard search.
RuleType: PrivacyByDeletion
RuleSource: GDPR
DataClassification:
  CustomerCompany:
    - CustomerCompanyName
    - CustomerCompanyCountry
ObjectFilter:
  CustomerCompany:
    CustomerCompanyName: someCompanyName
    WildcardSearch: 0
```

Delete customer company name and customer company country by customer company name with wildcard search:

```
---
RuleName: Delete customer company name and customer company country by
↳customer company name with wildcard search.
RuleSource: someRuleSource
RuleType: PrivacyByDeletion
DataClassification:
  CustomerCompany:
    - CustomerCompanyName
    - CustomerCompanyCountry
ObjectFilter:
  CustomerCompany:
    CustomerCompanyName: someCompanyName
    WildcardSearch: 1
```

Customer User Driver

The customer user driver provides the functionality to search and modify the information for customer users.

Possible data classifications:

```
- UserTitle
- UserFirstname
- UserLastname
- userEmail
- UserLogin
- UserComment
- UserCountry
- UserFax
- UserMobile
- UserCity
- UserPhone
- UserTitle
- UserStreet
- UserZip
- DynamicField_NameX
```

The driver supports dynamic fields for data classification. Dynamic fields will be identified by the prefix `DynamicField_` and the related field name.

Possible object filters:

```
- UserCity
- UserTitle
- UserFirstname
- UserPhone
- ValidID
- UserCountry
- UserLogin
- UserCustomerID
- UserLastname
- UserZip
- UserMobile
- userEmail
- UserFax
- WildcardSearch
- UserStreet
- UserComment
```

Object filter descriptions:

- `Limit`: Limits the number of search results.
- `CreateTime`: Searches for dates *greater than or equal to* (\geq) the given time.
- `Valid`: Searches for valid or invalid users. Possible values are 0 or 1.
- `WildcardSearch`: Affects all object filters, except `ValidID`. It will wrap the search term in wildcards, so they match all objects that contain the supplied value. For example, if this option is set to 1, filtering on `UserCustomerID` with `company` will in fact search for `*company*`.

Rule Configuration Examples

Here are some examples for rule configurations. These examples are valid YAML codes. You can copy these examples and modify them according to your needs.

Delete user first names and user last names by user first name with wildcard search:

```

---
RuleName: Delete user first names and user last names by user first name with_
↳wildcard search.
RuleType: PrivacyByDeletion
RuleSource: GDPR
DataClassification:
  CustomerUser:
    - UserFirstname
    - UserLastname
ObjectFilter:
  CustomerUser:
    UserFirstname: someFirstname
    WildcardSearch: 1

```

Anonymize user first names and user last names by user first name and without wildcard search:

```

---
RuleName: Anonymize user first names and user last names by user first name and_
↳without wildcard search.
RuleSource: someRuleSource
RuleType: PrivacyByAnonymization
DataClassification:
  CustomerUser:
    - UserFirstname
    - UserLastname
ObjectFilter:
  CustomerUser:
    UserFirstname: someFirstname
    WildcardSearch: 0

```

Delete user first names and user last names by user first name and user last name with wildcard search:

```

---
RuleName: Delete user first names and user last names by user first name and_
↳user last name with wildcard search.
RuleSource: someRuleSource
RuleType: PrivacyByDeletion
DataClassification:
  CustomerUser:
    - UserFirstname
    - UserLastname
ObjectFilter:
  CustomerUser:
    UserFirstname: someFirstname
    UserLastname: someLastname
    WildcardSearch: 1

```

Ticket Driver

The ticket driver provides the functionality to search and modify the information for tickets and related articles. Possible data classifications for tickets:

- Title
- CustomerUserID
- CustomerID
- DynamicField_NameX

Possible data classifications for articles:

- From
- To
- Cc
- Subject
- Body
- Attachments
- DynamicField_NameX

The driver supports dynamic fields for data classification. Dynamic fields will be identified by the prefix `DynamicField_` and the related field name.

The data classification supports history types. Since history types may vary (framework versions, framework updates, installed packages etc.), the driver will dynamically determine those types by name.

History types have to be prefixed with the term `History`. Please see the examples in the following listing:

- HistoryAddNote
- HistoryAddSMS
- HistoryArchiveFlagUpdate
- HistoryBounce
- HistoryCustomerUpdate
- HistoryEmailAgent
- HistoryEmailCustomer
- HistoryEmailResend
- HistoryEscalationResponseTimeNotifyBefore
- HistoryEscalationResponseTimeStart
- HistoryEscalationResponseTimeStop
- HistoryEscalationSolutionTimeNotifyBefore
- HistoryEscalationSolutionTimeStart
- HistoryEscalationSolutionTimeStop
- HistoryEscalationUpdateTimeNotifyBefore
- HistoryEscalationUpdateTimeStart
- HistoryEscalationUpdateTimeStop
- HistoryFollowUp
- HistoryForward
- HistoryLock
- HistoryLoopProtection
- HistoryMerged
- HistoryMisc
- HistoryMove
- HistoryNewTicket
- HistoryOwnerUpdate
- HistoryPhoneCallAgent
- HistoryPhoneCallCustomer
- HistoryPriorityUpdate
- HistoryRemove
- HistoryResponsibleUpdate

(continues on next page)

(continued from previous page)

- HistorySendAgentNotification
- HistorySendAnswer
- HistorySendAutoFollowUp
- HistorySendAutoReject
- HistorySendAutoReply
- HistorySendCustomerNotification
- HistoryServiceUpdate
- HistorySetPendingTime
- HistorySLAUpdate
- HistoryStateUpdate
- HistorySubscribe
- HistorySystemRequest
- HistoryTicketDynamicFieldUpdate
- HistoryTicketLinkAdd
- HistoryTicketLinkDelete
- HistoryTimeAccounting
- HistoryTitleUpdate
- HistoryTypeUpdate
- HistoryUnlock
- HistoryUnsubscribe
- HistoryWebRequestCustomer

All contents of the classified history types will be affected during executions.

If attachments are classified, every attachment of all matching articles or tickets will be affected during executions.

Warning: The ticket driver is used to search for tickets, even if the rule contains filters for article fields. If article fields are part of the data classification, all articles of the related, matching ticket will be processed!

The following fields can be used as search terms or filters for tickets and articles. Possible object filters:

- Limit
- TicketID
- TicketNumber
- Title
- Queues
- QueueIDs
- UseSubQueues
- Types
- TypeIDs
- States
- StateIDs
- StateType
- StateTypeIDs
- Priorities
- PriorityIDs
- Services
- ServiceIDs
- SLAs
- SLAIDs

(continues on next page)

(continued from previous page)

- Locks
- LockIDs
- OwnerIDs
- ResponsibleIDs
- WatchUserIDs
- CustomerID
- CustomerUserLogin
- CreatedUserIDs
- CreatedTypes
- CreatedTypeID
- CreatedPriorities
- CreatedPriorityIDs
- CreatedStates
- CreatedStateIDs
- CreatedQueues
- CreatedQueueIDs
- TicketFlag
- ArticleFlag
- MIMEBase_From
- MIMEBase_To
- MIMEBase_Cc
- MIMEBase_Subject
- MIMEBase_Body
- AttachmentName
- FullTextIndex
- ContentSearch
- ContentSearchPrefix
- ContentSearchSuffix
- ConditionInline
- ArticleCreateTimeOlderMinutes
- ArticleCreateTimeNewerMinutes
- ArticleCreateTimeNewerDate
- ArticleCreateTimeOlderDate
- TicketCreateTimeOlderMinutes
- TicketCreateTimeNewerMinutes
- TicketCreateTimeNewerDate
- TicketCreateTimeOlderDate
- TicketChangeTimeOlderMinutes
- TicketChangeTimeNewerMinutes
- TicketLastChangeTimeOlderMinutes
- TicketLastChangeTimeNewerMinutes
- TicketLastChangeTimeNewerDate
- TicketLastChangeTimeOlderDate
- TicketChangeTimeNewerDate
- TicketChangeTimeOlderDate
- TicketCloseTimeOlderMinutes
- TicketCloseTimeNewerMinutes
- TicketCloseTimeNewerDate
- TicketCloseTimeOlderDate
- TicketPendingTimeOlderMinutes
- TicketPendingTimeNewerMinutes
- TicketPendingTimeNewerDate

(continues on next page)

(continued from previous page)

```

- TicketPendingTimeOlderDate
- TicketEscalationTimeOlderMinutes
- TicketEscalationTimeNewerMinutes
- TicketEscalationTimeNewerDate
- TicketEscalationTimeOlderDate
- TicketEscalationUpdateTimeOlderMinutes
- TicketEscalationUpdateTimeNewerMinutes
- TicketEscalationUpdateTimeNewerDate
- TicketEscalationUpdateTimeOlderDate
- TicketEscalationResponseTimeOlderMinutes
- TicketEscalationResponseTimeNewerMinutes
- TicketEscalationResponseTimeNewerDate
- TicketEscalationResponseTimeOlderDate
- TicketEscalationSolutionTimeOlderMinutes
- TicketEscalationSolutionTimeNewerMinutes
- TicketEscalationSolutionTimeNewerDate
- TicketEscalationSolutionTimeOlderDate
- ArchiveFlags

```

All possible object filter parameters can be used to filter tickets and articles. Most of the attributes can be single strings or array references, like:

```

TicketNumber: 123546
TicketNumber:
  - 123546
  - 123666

```

```

Title: SomeText
Title:
  - SomeTest1
  - SomeTest2

```

```

States:
  - new
  - open
StateIDs:
  - 3
  - 4

```

The corresponding YAML code could look as follows:

```

RuleName: My Explanation Rule
RuleType: PrivacyByDeletion
RuleSource: GDPR
DataClassification:
  Ticket:
    - CustomerUserID
    - CustomerID
ObjectFilter:
  Ticket:
    Queue:
      - Junk

```

(continues on next page)

(continued from previous page)

- Raw
- Services:**
- Service A
- Service B

This rule would find all tickets, that are located in the queue *Junk* or *Raw* and which have the service *Service A* or *Service B* assigned. The fields `CustomerUserID` and `CustomerID` would be deleted.

There are several possible filter parameters, regarding relative times and dates, like:

- `ArticleCreateTimeOlderMinutes`
- `ArticleCreateTimeNewerMinutes`
- `ArticleCreateTimeNewerDate`
- `ArticleCreateTimeOlderDate`

A filter like `**TimeOlderMinutes*` means *older than X minutes*.

The following statement would mean: all tickets, that have a `CreateTime` older than one day (1440 minutes).

```
TicketCreateTimeOlderMinutes: 1440
```

The following statement would mean: all tickets, that have a `CreateTime` newer than one day (1440 minutes).

```
TicketCreateTimeNewerMinutes: 1440
```

This is principal valid for all filter parameters with this syntax.

For more descriptions about the single search parameters, check the `TicketSearch()` in [API reference](#).

Rule Configuration Examples

Here are some examples for rule configurations. These examples are valid YAML codes. You can copy these examples and modify them according to your needs.

Delete ticket titles by state names, that are older than one month:

```
---
RuleName: Delete ticket titles by state names, that are older than one month.
RuleSource: GDPR
RuleType: deletion
DataClassification:
  Ticket:
    - Title
ObjectFilter:
  Ticket:
    State:
      - new
      - open
  TicketCreateTimeOlderMinutes: 43200
```

Delete article subject and body by state names, that are located in specific queues:

```

---
RuleName: Delete article subject and body by state names, that are located in
↳specific queues.
RuleSource: GDPR
RuleType: deletion
DataClassification:
  Ticket:
    - Subject
    - Body
ObjectFilter:
  Ticket:
    State:
      - new
      - open
    Queues:
      - Postmaster
      - Misc

```

Pseudonymize customer user IDs for tickets, that are closed and archived:

```

---
RuleName: Pseudonymize customer user IDs for tickets, that are closed and
↳archived.
RuleSource: GDPR
RuleType: PrivacyByPseudonymization
DataClassification:
  Ticket:
    - CustomerUserID
ObjectFilter:
  Ticket:
    StateType:
      - Closed
    ArchiveFlags:
      - y

```

Anonymize customer IDs and some dynamic fields, that are closed, have certain services and are located in specific queues:

```

---
RuleName: Anonymize Customer IDs and some dynamic fields, that are closed,
↳have certain services and are located in specific queues.
RuleSource: GDPR
RuleType: PrivacyByAnonymization
DataClassification:
  Ticket:
    - CustomerID
    - DynamicField_SensitiveNames
    - DynamicField_SensitiveLocations
ObjectFilter:
  Ticket:
    StateType:
      - Closed
    Queue:

```

(continues on next page)

(continued from previous page)

- Special Queue A
- Junk
- Services:**
- Sensitive Customer Service
- VIP Customer Service

User Driver

The user driver provides the functionality to search and modify the information for users.

Possible data classifications:

- UserTitle
- UserFirstname
- UserLastname
- UserEmail
- UserMobile

Possible object filters:

- UserFirstname
- UserLastname
- UserLogin
- UserTitle
- CreateTime
- Valid
- Limit
- UserPreferences
- WildcardSearch

Object filter descriptions:

- **Limit:** Limits the number of search results.
- **CreateTime:** Searches for dates *greater than or equal to* (>=) the given time.
- **Valid:** Searches for valid or invalid users. Possible values are 0 or 1.
- **WildcardSearch:** Affects the object filters `UserFirstname`, `UserLastname`, `UserLogin` and `UserTitle`. It will wrap the search term in wildcards, so they match all objects that contain the supplied value. For example, if this option is set to 1, filtering on `UserLogin` with *agent* will in fact search for **agent**.
- **UserPreferences:** Array containing the user preferences like user email address as keys with certain search criteria as values (see YAML configuration examples).

Rule Configuration Examples

Here are some examples for rule configurations. These examples are valid YAML codes. You can copy these examples and modify them according to your needs.

Delete user first names by user first name:

```

---
RuleName: Delete user first names by user first name.
RuleSource: GDPR
RuleType: PrivacyByDeletion
DataClassification:
  User:
    - UserFirstname
ObjectFilter:
  User:
    UserFirstname: someFirstname

```

Delete user first names and user last names by user email:

```

---
RuleName: Delete user first names and user last names by user email.
RuleSource: GDPR
RuleType: PrivacyByDeletion
DataClassification:
  User:
    - UserFirstname
    - UserLastname
ObjectFilter:
  User:
    UserPreferences:
      UserEmail: someMail@example.com

```

Delete user first names and user last names with wildcard search:

```

---
RuleName: Delete user first names and user last names with wildcard search.
RuleSource: GDPR
RuleType: PrivacyByDeletion
DataClassification:
  User:
    - UserFirstname
    - UserLastname
ObjectFilter:
  User:
    UserFirstname: someFirstname
    WildcardSearch: 1

```

Delete user first names by user first name and create time, which are greater than or equal with the specified date:

```

---
RuleName: Delete user first names by user first name and create time, which are
↳greater than or equal with the specified date.
RuleSource: GDPR
RuleType: PrivacyByDeletion
DataClassification:
  User:
    - UserFirstname
ObjectFilter:

```

(continues on next page)

(continued from previous page)

User:

CreateTime: 2019-01-01

UserFirstname: someFirstname

17.2 Agent Interface

This package has no agent interface.

17.3 External Interface

This package has no external interface.

Delete Attachments

Enables the automatic deletion of ticket attachments that were closed in a definable time span.

Furthermore, file types can be defined as additional filter criteria. For example:

- .pdf
- .doc
- .xls
- etc.

Benefits

- Time saving through automatic deletions.
- Reduced administrative effort.

Target Groups

- Human resources
- Internal and external IT
- Finance
- Public authorities

Available in Service Package

- SILVER

Package Name in OTRS Package Manager

- OTRSDeleteAttachments

18.1 Administrator Interface

This package has no administrator interface.

18.1.1 System Configuration

The package is not working out-of-the-box, it must be configured by an administrator first.

1. Go to the *System Configuration* screen.
2. Select *OTRSDeleteAttachments* in the *Navigation* widget.
3. Navigate to *Core* → *Ticket* in the navigation tree.
4. Add some file MIME types to setting `OTRSDeleteAttachments::Filetypes` as key-value pairs. The key is the MIME type, the value is the deletion flag.
 - 0: This type of attachments are not to be deleted.
 - 1: This type of attachments are going to deleted.

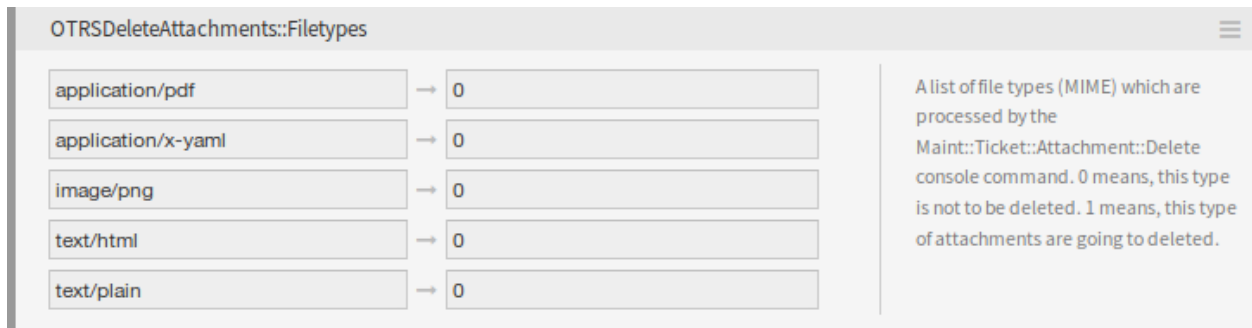


Fig. 1: File Types Setting

Automatic File Type Population

There is a console command `Maint::Ticket::Attachment::AddMIMETypes` to populate the system configuration setting automatically.

```
otrs> /opt/otrs/bin/otrs.Console.pl Maint::Ticket::Attachment::AddMIMETypes
Searching attachment types received the last 90 days...
|      Added undefined MIME type: application/x-yaml
|      Added undefined MIME type: image/png
|      Added undefined MIME type: text/html
|      Added undefined MIME type: application/pdf
|      Added undefined MIME type: text/plain
Updating SysConfig...
Done.
```

Note: The console command collects only MIME types of attachments, that are in closed tickets and received within the last 90 days.

Delete Attachments From Tickets

The attachment deletion is done by console command `Maint::Ticket::Attachment::Delete`. Check the following settings to customize the job:

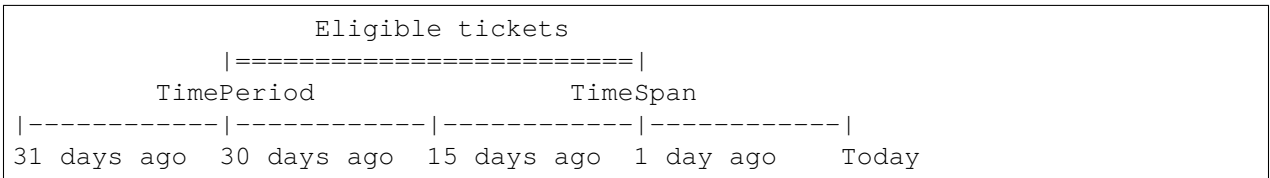
OTRSDeleteAttachments::Queues A list of queues (specified by their names) which are processed by the `Maint::Ticket::Attachment::Delete` console command.

Warning: If no queues are defined, all queues will be affected.

OTRSDeleteAttachments::TimePeriod The period (in days) which should be analyzed, starting from the time when the ticket was closed, backwards.

OTRSDeleteAttachments::TimeSpan The minimum time span (in days) between the current time and the time when the ticket was closed, to delete the attachments of a ticket.

With a `TimeSpan` value of `1` and `TimePeriod` value of `30` means that the eligible tickets to delete their attachments are the ones that has been closed between 30 days ago and yesterday.



See also:

The console command `Maint::Ticket::Attachment::Delete` has some other options. Run the command with the `--help` option to see the possibilities.

```
otrs> /opt/otrs/bin/otrs.Console.pl Maint::Ticket::Attachment::Delete --help
```

Automatic Execution

Updating the file types and deleting the attachments can be run in the background by the OTRS daemon. Enable the following settings to run the jobs in a regular basis:

Daemon::SchedulerCronTaskManager::Task###OTRSDeleteAttachments Delete attachments from tickets.

Daemon::SchedulerCronTaskManager::Task###OTRSDeleteAttachments_AddMIMETypes Populate system configuration MIME types for attachment deletion.

Note: Do not forget to restart the OTRS daemon after the system configuration is changed.

18.2 Agent Interface

This package has no agent interface.

18.3 External Interface

This package has no external interface.

Dynamic Field Attachment

Increase the information flow in your tickets and articles significantly with this feature add-on. Add dynamic fields of the type attachment, which can store one or many files. Define the maximum size and the maximum amount of attachments. Share pictures, PDF documents, Excel lists and basically any file you want within dynamic fields of articles and tickets.

Benefits

- Add attachments wherever you want in the system.
- Have control over how many and which size of attachments your agent should use.
- Possibility to use **OTRS** as your document management system.

Target Groups

- Internal and external IT service provider
- Government departments
- Human resources
- Sales
- Customer service

Available in Service Package

- GOLD

Package Name in OTRS Package Manager

- OTRSDynamicFieldAttachment

19.1 Administrator Interface

This package contains a configuration interface to create dynamic fields of the type *Attachment*. These fields can store files in a dynamic field and display them in the different views.

19.1.1 Processes & Automation

This package makes it possible to create dynamic fields of the type *Attachment*, which can store files in tickets and articles.

Dynamic Fields

After installation of the package a new dynamic field type *Attachment* will be available for tickets and articles.

This dynamic field can be created the same way as default dynamic fields are created. For this navigate to the *Dynamic Fields* module of the *Processes & Automation* group in the administrator interface. In this screen you can select the *Attachment* field from the drop-down lists on the left side.

See also:

The usage of dynamic fields and the general dynamic field settings are described in the [Dynamic Fields](#) chapter of the administrator manual.

Dynamic Field Settings

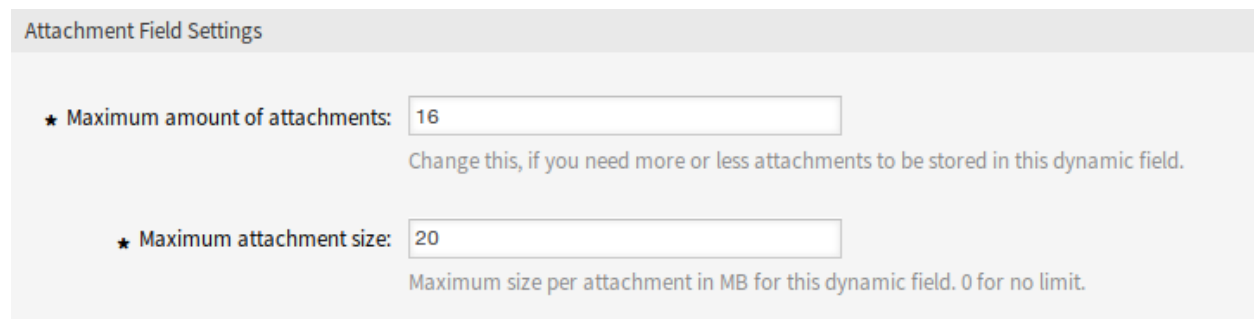
The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

Attachment Dynamic Field Settings

Attachment dynamic field is used to store attachments for tickets and articles.

Article dynamic fields can hold attachments for each ticket article. Attachments stored in an article dynamic field of type attachment can just be downloaded, but not deleted or changed.

Ticket dynamic fields can hold attachments for each ticket. Attachments stored in a ticket dynamic field of type attachment are stored once for each ticket and can be downloaded and deleted.



Attachment Field Settings

★ Maximum amount of attachments:
Change this, if you need more or less attachments to be stored in this dynamic field.

★ Maximum attachment size:
Maximum size per attachment in MB for this dynamic field. 0 for no limit.

Fig. 1: Attachment Dynamic Field Settings

Maximum amount of attachments * This setting holds the amount of files that can be stored in the dynamic field of type attachment per ticket or article. Increasing this value enables the agents to add more files. Reducing this amount does not delete attachments already stored in dynamic fields of type attachment, but reduces the possibility for adding files up to the configured maximum.

Maximum attachment size * This setting holds the maximum file size in megabyte each uploaded file can have. If an agent is uploading a file greater than the here configured file size, the file is accepted initially. As soon as the upload is done, the size is checked and the file gets deleted as well as the user informed that the file was not stored because it exceeded the maximum attachment size limit.

Note: Do not forget to add the new dynamic field to ticket view screens.

Generic Agent

Dynamic fields of type attachment could be used only in *Select Tickets* section in generic agent. However, it is not updatable by generic agent and this type of dynamic fields are not listed in *Update/Add Ticket Attributes* section.

19.2 Agent Interface

This package has no agent interface, but the dynamic field of type attachment can be added to many screens.

To upload files to dynamic field of type attachment:

1. Make sure that the dynamic field of type attachment is added to the desired screen.
2. Click on the uploading area or just dropping the files there. The file gets uploaded immediately to the server and a new upload field will be displayed as long as the configured maximum amount of attachments has not been reached.
3. Click on the *Send* button. If a file has exceeded the maximum attachment size, a message will appear warning the agent that the file exceeds the maximum attachment size. In this case the file will not be uploaded and the ticket will be created without problem.

Documents

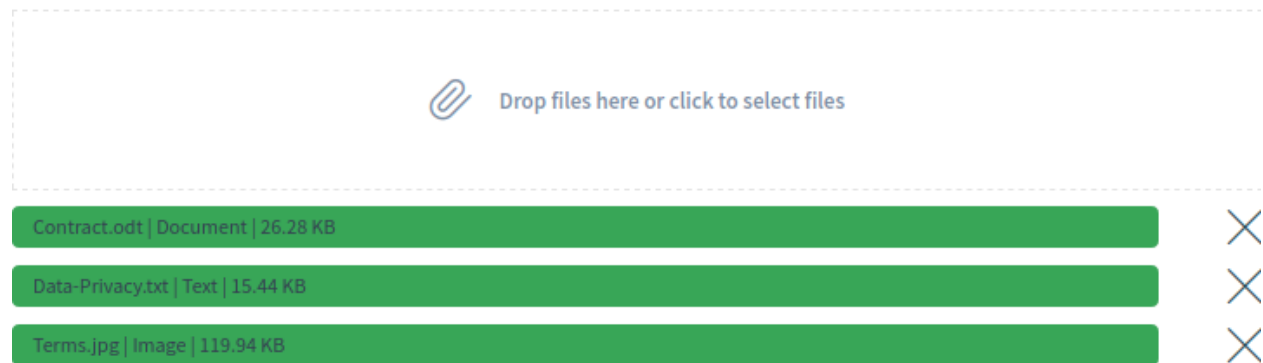


Fig. 2: Dynamic Field Attachment in Action Form

The dynamic field of type attachment is displayed in the ticket detail view or in the header of the article, if it is added to the screens and at least one file has been uploaded.

To download a file from a ticket dynamic field of type attachment:

1. Find the dynamic field in the *Properties* widget of the ticket detail view.
2. Click on the download icon next to the filename to download the file.

The widget can be expanded to look similar to the *Attachments* widget.

If an image file is uploaded, it is possible to see a preview of the image.

To download a file from an article dynamic field of type attachment:

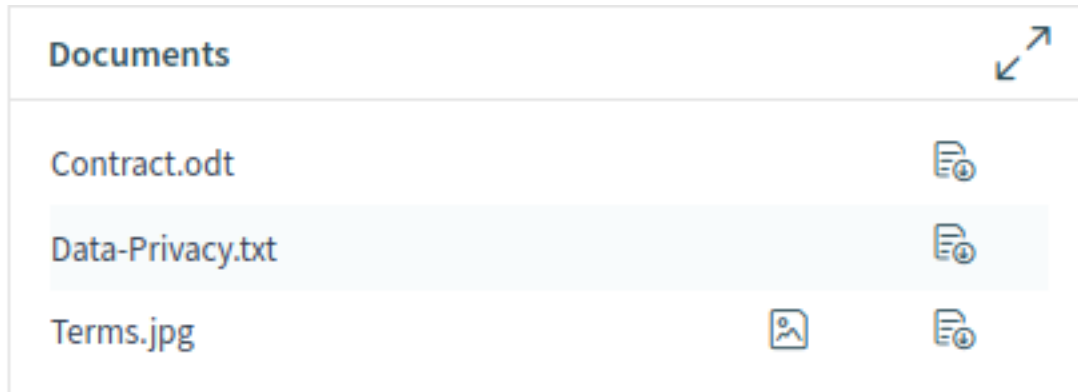


Fig. 3: Dynamic Field Attachment in Ticket Detail View

Documents				
Type	Filename	Filesize	Preview	Download
	Contract.odt	26.28 KB		
	Data-Privacy.txt	15.44 KB		
	Terms.jpg	119.94 KB		

Fig. 4: Expanded Dynamic Field Attachment in Ticket Detail View

1. Select an article, that has a dynamic field of type attachment with at least one uploaded file.
2. Expand the article.
3. Find the attachments in the article header.
4. Click on a filename to download the file.

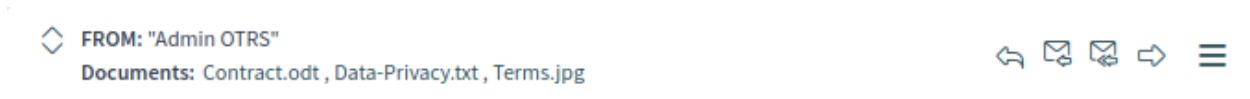


Fig. 5: Dynamic Field Attachment in Article Header

If a ticket dynamic field of type attachment has been configured to be displayed in a ticket action form, it is possible to delete the uploaded files and add new files.

To delete a file from a ticket dynamic field of type attachment:

1. Make sure that the dynamic field of type attachment is added to the *Change Free Fields* action.
2. Click on the X icon in the last column of an attachment. The attachment will be deleted without confirmation!

Note: It is not possible to delete files from dynamic field of type attachment created for article object.

Dynamic field of type attachment supports searching for the file names. To search for a file name of a dynamic field of type attachment, add the dynamic field to the search engine.

19.3 External Interface

This package has no external interface, but the dynamic field of type attachment can be added to many screens.

Dynamic Field Calculation

This feature add-on makes it possible to create dynamic fields of the type *Calculation*, which can perform simple mathematical operations.

They can be used when a sum is required, such as with invoices, reporting or other number-related data. Administrators can run calculations in numeric fields if they want to add or divide values from one numeric field into another numeric field.

Benefits

- Perform calculations in numeric fields.
- Save time by easily adding or multiplying values in a numeric field.

Target Groups

- Finance and accounting
- Project management
- Security management

Available in Service Package

- GOLD

Package Name in OTRS Package Manager

- OTRSDynamicFieldCalculation

20.1 Administrator Interface

This package contains a configuration interface to create dynamic fields of the type *Calculation*. These fields can perform simple mathematical operations.

20.1.1 Processes & Automation

This package makes it possible to create dynamic fields of the type *Calculation*, which can perform simple mathematical operations.

Dynamic Fields

After installation of the package a new dynamic field type *Calculation* will be available for tickets and articles.

This dynamic field can be created the same way as default dynamic fields are created. For this navigate to the *Dynamic Fields* module of the *Processes & Automation* group in the administrator interface. In this screen you can select the *Calculation* field from the drop-down lists on the left side.

See also:

The usage of dynamic fields and the general dynamic field settings are described in the [Dynamic Fields](#) chapter of the administrator manual.

Dynamic Field Settings

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

Calculation Dynamic Field Settings

Calculation dynamic field is used to perform simple mathematical operations.

Calculation Field Settings

Calculation formula:

Note: Simple mathematical operations can be executed. Basic arithmetic operations (+, -, *, /) are allowed.

<OTRS_TICKET_DynamicField_Name1> + <OTRS_TICKET_DynamicField_Name2>
 <OTRS_TICKET_DynamicField_Name1> * <OTRS_TICKET_AccountedTime> +
 <OTRS_TICKET_DynamicField_Name2>
 (<OTRS_TICKET_DynamicField_Name1> + <OTRS_TICKET_DynamicField_Name2>) *
 <OTRS_TICKET_AccountedTime>

Fig. 1: Calculation Dynamic Field Settings

Calculation formula This setting is used to add a calculation formula. Simple mathematical operations are possible. The default value is zero.

Note: Do not forget to add the new dynamic field to ticket view screens.

20.2 Agent Interface

This package has no agent interface, but the dynamic field of type calculation can be added to many screens.

20.3 External Interface

This package has no external interface.

Dynamic Field CI

This feature add-on makes it possible to show configuration items from the OTRS CMDB in dynamic fields in ticket screens in the agent and external interface. This speeds up the ticket creation process for agents and customer users by enabling the use of filters so that only those configuration items that are relevant for the customer company are shown and can be selected in the ticket screens. In addition, the configuration item search function can also be used directly in the ticket. Filters can be defined, e.g. based on:

- Configuration item class
- Usage status
- Incident status
- Customer ID

In addition, the following configurations are possible:

- Automatic linking and type of linking between configuration item and ticket.
- How and in which ticket screens the configuration item should be shown (e.g. as a drop-down list or a tree view).
- Visualization in the external or agent interface.

Benefits

- Customer-based visualization and selection of configuration items when creating a ticket in the external interface.
- Faster ticket creation thanks to the use of filters to choose which configuration items should be shown, e.g. according to class, usage status, incident status, etc.
- Maximum flexibility through use in different ticket screens for customer users and/or agents.
- Rapid search for configuration items when creating a ticket.
- Useful for statistics and automatic ticket notifications.

Target Groups

- Customers of companies that offer services for different devices or products

- Customer service
- Internal and external IT
- Facility management
- Sales

Available in Service Package

- GOLD

Package Name in OTRS Package Manager

- OTRSDynamicFieldCI

Note: This feature add-on requires the *Configuration Management* feature.

21.1 Administrator Interface

This package contains a configuration interface to create dynamic fields of the type *Configuration item*. These fields can store configuration items in a dynamic field and display them in the different views.

21.1.1 Processes & Automation

This package makes it possible to create dynamic fields of the type *Configuration item*, which can store configuration items in tickets.

Dynamic Fields

After installation of the package a new dynamic field type *Configuration item* will be available for tickets.

This dynamic field can be created the same way as default dynamic fields are created. For this navigate to the *Dynamic Fields* module of the *Processes & Automation* group in the administrator interface. In this screen you can select the *Configuration Item* field from the drop-down lists on the left side.

See also:

The usage of dynamic fields and the general dynamic field settings are described in the [Dynamic Fields](#) chapter of the administrator manual.

Dynamic Field Settings

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

Configuration Item Dynamic Field Settings

Dynamic field of type configuration item is used to store configuration items for tickets.

Configuration Item Field Settings

Filter Options

Class:

Filter for configuration item classes.

Deployment State:

Filter for configuration item deployment states.

Incident State:

Filter for configuration item incident states.

Restriction Options

▼ Restriction by Customer ID

Restriction by Interface:

Select if the available configuration items should be restricted by customer ID in agent and/or external interface.

Configuration Item Field:

The name of the configuration item field containing the customer ID.

Value Lookup for Customer ID

Attribute: Select if the readable value for the customer ID field should be used for restriction.

AND

▼ Restriction by Customer User ID

Restriction by Interface:

Select if the available configuration items should be restricted by customer user ID in agent and/or external interface.

Configuration Item Field:

The name of the configuration item field containing the customer user ID.

Value Lookup for Customer User ID

Attribute: Select if the readable value for the customer user ID field should be used for restriction.

AND

▼ Restriction by Attribute

Restriction by Other Attribute:

Select if the available configuration items should be restricted by an other attribute.

Filter Options

Class Filter for configuration item classes to narrow down the list of possible values.

Deployment State Filter for configuration item deployment states to narrow down the list of possible values.

Incident State Filter for configuration item incident states to narrow down the list of possible values.

Restriction Options

Restriction by Interface Select if the available configuration items should be restricted by customer ID, by customer user ID or by any other attribute in agent and/or external interface.

Configuration Item Field The name of the configuration item field containing the customer ID, the customer user ID or any other attribute. Usually this is the owner field.

Value Lookup for Customer ID, Customer User ID or Any Attribute Select if the readable value for the customer ID, the customer user ID or any other attribute field should be used for restriction.

Note: If the filters are changed after the configuration item and the ticket has been linked, the link will not be automatically updated.

Link Options

Link Type Type of automatically created links for configuration items. *None* disables the automatic linking.

Link Direction If a *Link Type* is selected, choose the link direction from the point of ticket.

Display Options

Multiselect Activate this option to allow multiple selection of values.

Add Empty Value Activate this option to create an empty selectable value.

Tree View Activate this option to display values as a tree.

Translatable Values If you activate this option the values will be translated to the user defined language.

Note: You need to add the translations manually into the language translation files.

Autocompletion Activate this option to enable autocompletion for the values.

Advanced Options

Follow CMDB Links If you activate this option CMDB items linked to the items set in the *Follow Dynamic Field* will be displayed as selectable options for this field.

CMDB Link Settings Define elements to show for this field if no items are selected in the *Follow Dynamic Field*. Possible values:

- Does not show any items
- Show regular items

Follow Dynamic Field Dynamic field to follow CMDB links from. This option only works if the setting *Restriction by Customer ID* or *Restriction by Customer User ID* is deactivated.

Follow Dynamic Field Attribute The name of the configuration item field containing the dynamic field value.

Note: Do not forget to add the new dynamic field to ticket view screens.

21.2 Agent Interface

This package has no agent interface, but the dynamic field of type configuration item can be added to many screens.

21.2.1 Automatic Linking and Unlinking

When configuration items are added to or removed from tickets via the new dynamic field back end, links for the respective configuration items can be added or removed automatically. If this is configured, links will be compared and links will be added and/or removed, if necessary.

Warning: In order to prevent inconsistencies between the dynamic field content and links please do not add or remove links of the configured type and direction manually via the link mechanism in the ticket detail view. Those changes are not synchronized back to the dynamic field content.

See also:

The dynamic field configuration decides if linking happens and what link type and direction will be used. See the [Dynamic Fields](#) in the administrator interface.

21.3 External Interface

This package has no external interface.

Dynamic Field Database

Integrate data from any number of external databases, such as an ERP, CRM, tracking or other ticket systems, and have it displayed in dynamic fields. These records can then be searched and pre-filtered with additional masks.

Benefits

- Time saved when searching, filtering (e.g. according to customer data) and overwriting ticket data.
- All data is centrally accessible in **OTRS**.
- Automatic transfer of additional data to the record.

Target Groups

- Complaint management
- Customer service
- Document management
- Facility management
- Human resources
- IT
- IT service
- Project management
- Security management
- Service providers
- Supply-chain management

Available in Service Package

- SILVER

Package Name in OTRS Package Manager

- OTRSDynamicFieldDatabase

22.1 Administrator Interface

This package contains a configuration interface to create dynamic fields of the type *Database*. These fields can store data from external databases in a dynamic field and display them in the different views. Such connected data sets can be searched and filtered using additional masks.

22.1.1 Processes & Automation

This package makes it possible to create dynamic fields of the type *Database*, which can store data from external databases in tickets.

Dynamic Fields

After installation of the package a new dynamic field type *Database* will be available for tickets.

This dynamic field can be created the same way as default dynamic fields are created. For this navigate to the *Dynamic Fields* module of the *Processes & Automation* group in the administrator interface. In this screen you can select the *Database* field from the drop-down lists on the left side.

See also:

The usage of dynamic fields and the general dynamic field settings are described in the [Dynamic Fields](#) chapter of the administrator manual.

Dynamic Field Settings

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

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

Database Dynamic Field Settings

Dynamic field of type database is used to store database records for tickets.

Possible values The possible values will fill up the *Identifier* field below automatically, which defines the value that will be stored in the dynamic field. Possible values can be created as much as needed (or at least as many table columns as the database table has). The possible values defines the database columns to search in. It is possible to set the column name, a description (label) the field should have, the needed data type and if the field should be a search or list field.

Name * The exact name of the database column which will be requested through the database queries.

Label * The label of the field which will be displayed in the detailed search.

Datatype * The data type which will be stored in the dynamic field. Possible values: *Date*, *Integer* or *Text*.

Database Field Settings

Possible values:

★ Name:

★ Label:

★ Datatype:

Filter:

Search field

List field

Add value:

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.

★ Type:

★ Server:

Port:

★ Database:

★ Table / View:

★ User:

★ Password:

★ Identifier:

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

Multiselect:

Cache TTL:

Search Prefix:

Search Suffix:

Result Limit:

Case Sensitive:

Fig. 1: Database Dynamic Field Settings

Filter With the filter field, it is possible to choose a ticket attribute or a dynamic field as a filter for the related column. If the dynamic field is bound to a related ticket, the attributes will be used for the filter mechanism, otherwise the filters will be ignored. If the filter will be configured to a table column, only search results matching to the search term and the related ticket attribute on exactly the configured column will be displayed.

Searchfield Indicates if a field should be included in the search requests.

Listfield Indicates if a field should be displayed in the results.

Add value Click on the + icon to add a new section to *Possible values*.

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.

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.

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

SID This option is only available for Oracle connections and will be shown or hidden automatically. Within this option you have to enter the SID of your Oracle connection.

Driver This option is only available for ODBC connections and will be shown or hidden automatically. Within this option you have to enter in the host system previously configured ODBC driver to connect to the desired MSSQL database.

Server * The database host (host name or IP address).

Port The port of the database server.

Database * Defines the desired target database of the DBMS. This database will be used for queries.

Table/View * This table or view will be used for the queries.

User * The username for the database connection.

Password * The user password for the database connection.

Identifier * This select box will be automatically filled through *Possible values*. This field represents the value which will be stored in the dynamic field. Must be unique column from the table entered in *Table/View*.

Multiselect If this field is selected, it will be possible to store more than one value to the dynamic field. Those values will be stored comma separated.

Cache TTL This value defines the period of validity of the database cache in seconds. Equal queries to the database will be answered through the cache (local file system) within this period instead of asking the database again.

Search Prefix This value will be put in the front of every search term while using the auto-completion to search the database. Wildcard characters are supported as well. The search prefix will be ignored during the detailed search, but it is still possible to use wildcard characters in those masks.

Search Suffix This value will be put in the end of every search term while using the auto-completion to search the database. Wildcard characters are supported as well. The search suffix will be ignored during the detailed search, but it is still possible to use wildcard characters in those masks.

Result Limit The entered integer value defines the maximum amount of allowed results during a database search. This includes the auto-completion search as well as the detailed search.

Case Sensitive If this field is selected, case-sensitivity will take effect on searches.

Note: Do not forget to add the new dynamic field to ticket view screens.

Storage of Historical Data

This module offers a functionality to store historical data. For this to work it is necessary to activate and set the settings in the system configuration available in *Core* → *DynamicFieldDatabase* navigation path.

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

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

22.2 Agent Interface

This package has no agent interface, but the dynamic field of type database can be added to many screens.

22.2.1 Searching and Saving Data Sets

After the created dynamic fields are activated in some screens (like *New Phone Ticket*, *New Email Ticket*) a new text field appears with the name, the dynamic field got in the configuration. In this field it is possible to input search terms and therefore execute a search over all configured database fields. Otherwise click on the *Detailed search* link and start a detailed search in which the fields to search in are selected explicitly. Wildcard character `*` is allowed in every single field.

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

Auto-completion

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

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

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

Detailed Search

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

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

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

22.3 External Interface

This package has no external interface, but the dynamic field of type database can be added to many screens.

The features added by this package are the same as described in the [Agent Interface](#).

Dynamic Field Web Service

With the help of this dynamic field, data from third-party systems can be displayed without custom development.

Benefits

- Saves time when processing tickets since updating third-party systems is not necessary.
- Data management does not have to take place within **OTRS**.
- Execute arbitrary complex mappings through the XSLT mapping module.

Target Groups

- Complaint management
- Customer service
- Development
- Document management
- E-commerce
- IT
- IT service
- Project management
- Security management

Available in Service Package

- SILVER

Package Name in OTRS Package Manager

- OTRSDynamicFieldWebService

23.1 Administrator Interface

This feature implements a generic dynamic field type *Web Service* that gathers its selectable options from an external system using a web service.

The response from the external system defines the possible options to be displayed, and they could vary depending on the data that is sent in the request. Normally when a field is changed in a screen (e. g. the ticket priority in the *New Phone Ticket* screen) the values of other fields could be updated. That is the case with this type of dynamic fields, as they could also include all screen field values in the request and the remote server could potentially return completely different values depending on input.

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

23.1.1 Processes & Automation

This package makes it possible to create dynamic fields of the type *Web Service*, which can store data from external systems in tickets.

Dynamic Fields

After installation of the package a new dynamic field type *Web Service* will be available for tickets.

This dynamic field can be created the same way as default dynamic fields are created. For this navigate to the *Dynamic Fields* module of the *Processes & Automation* group in the administrator interface. In this screen you can select the *Web Service* field from the drop-down lists on the left side.

See also:

The usage of dynamic fields and the general dynamic field settings are described in the [Dynamic Fields](#) chapter of the administrator manual.

Dynamic Field Settings

It is necessary to have an already working web service before creating a new dynamic field, that uses such web services.

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

Web Service Dynamic Field Settings

Dynamic field of type web service is used to store data from external systems for tickets.

Web service * The configured web service whose invokers will be triggered when a dynamic field is displayed.

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

Multiselect A drop-down menu to determine if the displayed dynamic field should act as a multiselect field instead of a drop-down field.

Web Service Field Settings

★ Web service:
The web service to be executed for possible values.

★ Invoker:
The invoker to be used to perform requests (invoker needs to be of type 'Generic::PassThrough').

Multiselect:
Activate this option to allow multiselect on results.

Cache TTL:
Cache time to live (in minutes), to save the retrieved possible values.

Add empty value:
Activate this option to create an empty selectable value.

Tree View:
Activate this option to display values as a tree.

Translatable values:
If you activate this option the values will be translated to the user defined language.
Note: You need to add the translations manually into the language translation files.

Show link:
Here you can specify an optional HTTP link for the field value displayed in overviews and detail views. Optional HTTP link works only for single-select fields.
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. 1: Web Service Dynamic Field Settings

Cache TTL A cache time to live value, that contains a value (in minutes). If the value is 0 or empty, no caching will be active. This cache is to prevent unnecessary requests to the remote server using the same values.

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

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

Translatable values 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. Optional HTTP link works only for single-select fields. If special characters (&, @, :, /, etc.) should not be encoded, use `url` instead of `uri` filter.

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.

Note: Do not forget to add the new dynamic field to ticket view screens.

Web Services

To create a new dynamic field of type *Web Service* it is necessary to have an already working web service. It requires to have at least one invoker of the type `Generic::PassThrough`. This invoker will be called to fetch the data from the remote server. The original data that it is sent in a request is similar to the following example.

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

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

Ticket This section (or another source object, e. g. `CustomerUser`) contains the attributes of the object where the dynamic field belongs.

For example in the *New Phone Ticket* screen the section is empty as the ticket is not created yet, but in the *Change Free Fields* screen it contains the information of the current ticket.

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

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

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

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:transform
  xmlns:xsl="https://www.w3.org/1999/XSL/Transform"
  xmlns:date="https://exsult.org/dates-and-times"
  version="1.0"
  extension-element-prefixes="date">

  <xsl:output method="xml" encoding="utf-8" indent="yes" />

  <!-- Don't return unmatched tags -->
  <xsl:template match="text()" />

  <!-- Remove empty elements -->
  <xsl:template match="*[not(node())]" />

  <!-- Root template -->
  <xsl:template match="/">
    <RootElement>
      <UserLogin>someuser</UserLogin>
      <Password>somepassword</Password>
      <TicketID>1</TicketID>
    </RootElement>
  </xsl:template>
</xsl:transform>
```

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

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

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:transform
  xmlns:xsl="https://www.w3.org/1999/XSL/Transform"
  xmlns:date="https://exsult.org/dates-and-times"
  version="1.0"
  extension-element-prefixes="date">
```

(continues on next page)

(continued from previous page)

```

<xsl:output method="xml" encoding="utf-8" indent="yes" />

<!-- Don't return unmatched tags -->
<xsl:template match="text()" />

<!-- Remove empty elements -->
<xsl:template match="*[not(node())]" />

<!-- Root template -->
<xsl:template match="/">
  <RootElement>
    <xsl:apply-templates />
  </RootElement>
</xsl:template>

<xsl:template match="*/Ticket">
  <PossibleValue>
    <Key>State</Key>
    <Value>
      <xsl:value-of select="*/Ticket/State" />
    </Value>
  </PossibleValue>
  <PossibleValue>
    <Key>Queue</Key>
    <Value>
      <xsl:value-of select="*/Ticket/Queue" />
    </Value>
  </PossibleValue>
</xsl:template>
</xsl:transform>

```

The following web service definition (importable YAML file) can be used for testing the field, but the endpoint must be adapted to match current system. This web service acts as requester and provider and it always returns the state and queue from TicketID 1, as possible values to the field.

Note: This example should not be used in conjunction with the development web server.

```

---
Debugger:
  DebugThreshold: debug
  TestMode: '0'
Description: Dynamic Field Web Service Test
FrameworkVersion: 7.0.x git
Provider:
  ErrorHandling: {}
  ErrorHandlingPriority: []
  Operation:
    TicketGet:
      Description: ''

```

(continues on next page)

(continued from previous page)

```

IncludeTicketData: ''
MappingInbound: {}
MappingOutbound: {}
Type: Ticket::TicketGet
Transport:
  Config:
    AdditionalHeaders: ~
    MaxLength: '1000000000'
    NameSpace: https://www.otrs.org/TicketConnector/
    RequestNameFreeText: ''
    RequestNameScheme: Plain
    ResponseNameFreeText: ''
    ResponseNameScheme: Response
  Type: HTTP::SOAP
RemoteSystem: ''
Requester:
  ErrorHandling: {}
  ErrorHandlingPriority: []
  Invoker:
    TicketGet:
      Description: Get possible values from the other side.
      Events: []
      MappingInbound:
        Config:
          Template: |-
            <?xml version="1.0" encoding="UTF-8"?>
            <!--
            Copyright (C) 2001-2019 OTRS AG, https://otrs.com/
            This software comes with ABSOLUTELY NO WARRANTY. For details,
            ↪see
            the enclosed file COPYING for license information (GPL). If you
            did not receive this file, see https://www.gnu.org/licenses/gpl.
            ↪txt.
            -->

            <!-- DOCUMENTATION

            * Example XML Input *
            <RootElement>
                ...
            </RootElement>

            * Example XML Output *
            <RootElement>
                <PossibleValues>
                    <Key>???

```

(continues on next page)

(continued from previous page)

```

        </PossibleValues>
        ...
    </RootElement>

-->

<xsl:transform
  xmlns:xsl="https://www.w3.org/1999/XSL/Transform"
  xmlns:date="https://exslt.org/dates-and-times"
  version="1.0"
  extension-element-prefixes="date">

  <xsl:output method="xml" encoding="utf-8" indent="yes" />

  <!-- Don't return unmatched tags -->
  <xsl:template match="text()" />

  <!-- Remove empty elements -->
  <xsl:template match="*[not(node())]" />

  <!-- Root template -->
  <xsl:template match="/">
    <RootElement>
      <xsl:apply-templates />
    </RootElement>
  </xsl:template>

  <xsl:template match="*/Ticket">
    <PossibleValue>
      <Key>State</Key>
      <Value><xsl:value-of select="*/Ticket/State" /></
→Value>
    </PossibleValue>
    <PossibleValue>
      <Key>Queue</Key>
      <Value><xsl:value-of select="*/Ticket/Queue" /></
→Value>
    </PossibleValue>
  </xsl:template>

</xsl:transform>

```

Type: XSLT

MappingOutbound:

Config:

Template: |-

```

<?xml version="1.0" encoding="UTF-8"?>
<!--
Copyright (C) 2001-2019 OTRS AG, https://otrs.com/

This software comes with ABSOLUTELY NO WARRANTY. For details,
→see

```

(continues on next page)

(continued from previous page)

```

the enclosed file COPYING for license information (GPL). If you
did not receive this file, see https://www.gnu.org/licenses/gpl.
->txt.
-->

<!-- DOCUMENTATION

* Example XML Input *
<RootElement>
    ...
</RootElement>

* Example XML Output *
<RootElement>
    <PossibleValues>
        <Key>???

```

Type: XSLT

(continues on next page)

(continued from previous page)

```
Type: Generic::PassThrough
Transport:
Config:
  Encoding: ''
  Endpoint: https://localhost/otrs/nph-genericinterface.pl/Webservice/
↳GenericConfigItemConnectorSOAP
  Namespace: https://www.otrs.org/TicketConnector/
  RequestNameFreeText: ''
  RequestNameScheme: Plain
  ResponseNameFreeText: ''
  ResponseNameScheme: Response
  SOAPAction: Yes
  SOAPActionSeparator: '#'
  SSL:
    SSLProxy: ''
    SSLProxyPassword: ''
    SSLProxyUser: ''
  Type: HTTP::SOAP
  UseMappedData: '1'
```

23.2 Agent Interface

This package has no agent interface.

23.3 External Interface

This package has no external interface.

Dynamic Sender Addresses

The sender address of your tickets in the standard version of **OTRS** is by default a queue-associated email address. In order to configure the sender address this feature add-on uses a special dynamic field. When replying or forwarding tickets and event-based notifications, **OTRS** automatically sets the new sender address. When the dynamic field is blank, the queue-associated email address is used.

In addition, you can also use this feature add-on to edit tickets from all over the world and use country-specific sender addresses. By grouping the countries and email addresses **OTRS** can automatically add the correct return address for each country.

Benefits

- Worldwide service with regional sender addresses.
- Easier identification of incoming emails for your customers.
- Improved customer loyalty.
- Simplified teamwork with partners or outsourced divisions.

Target Groups

- Call centers
- Customer service
- Sales
- Marketing
- Procurement

Available in Service Package

- GOLD

Package Name in OTRS Package Manager

- OTRSDynamicSenderAddresses

24.1 Administrator Interface

This package allows to base the sender address of outgoing emails on a dynamic field. This can be used for replies, forwards and event based notifications.

24.1.1 Communication & Notifications

After installation of the package, a new administrator module allows to add and modify values to be mapped from the configured dynamic field values into system email addresses. For each added value a valid system address can be selected. This mapping will be checked against dynamic field values in order to determine the sender address.

Sender Address Mapping

Use this screen to map the values contained in the configured dynamic field into system email addresses. The sender address mapping screen is available in the *Sender Address Mapping* module of the *Communication & Notifications* group.

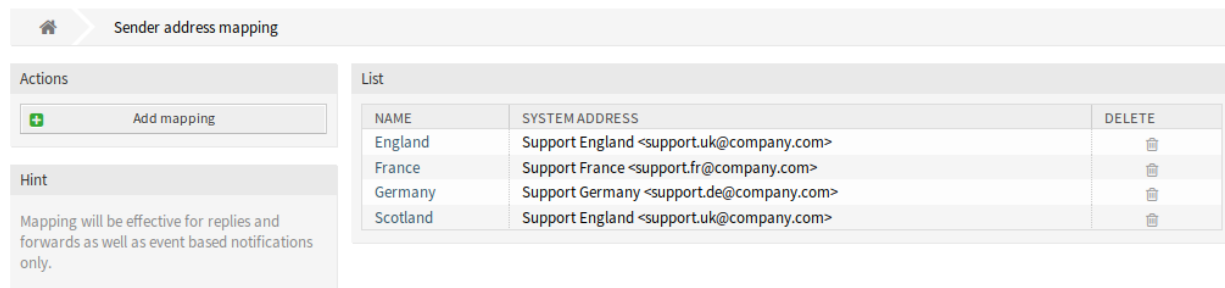


Fig. 1: Sender Address Mapping Management Screen

Map Email Addresses

Note: The mapping is based on the dynamic field set in setting `DynamicSenderAddresses###FieldName`. The dynamic field has to be created manually.

To add a mapping:

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

To edit a mapping:

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

To delete a mapping:

Add mapping

★ Name:

★ System address:

or

Fig. 2: Add Mapping Screen

Edit mapping

★ Name:

★ System address:

or or

Fig. 3: Edit Mapping Screen

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

List

NAME	SYSTEM ADDRESS	DELETE
England	Support England <support.uk@company.com>	
France	Support France <support.fr@company.com>	
Germany	Support Germany <support.de@company.com>	
Scotland	Support England <support.uk@company.com>	

Fig. 4: Delete Mapping Screen

See also:

This administration screen reads and writes the system configuration setting `DynamicSenderAddresses###Mapping`, where the key is the name of the mapping and the value is the internal ID of the system email address.

Warning: Do not change the setting manually. Use the administrator interface module instead.

Mapping 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.

Country	Value
England	3
France	4
Germany	2
Scotland	3

Mapping from dynamic field value to sender address. Do not change manually - use admin frontend module!

Fig. 5: DynamicSenderAddresses###Mapping Setting

System address * Select a system email address configured in *Email Addresses* module in the administrator interface.

24.1.2 Processes & Automation

A generic agent module is provided to have ticket values copied to a dynamic field to use this field as base for the sender address.

Generic Agent

The ticket service should be the base for the sender address, but only dynamic field values are supported by this feature.

In order to allow using the service, create a new dynamic field and use a generic agent job. The helper module will be added in order to automatically fill the new dynamic field with the service name.

To fill the dynamic field value with the service name:

1. Create the dynamic field *Service Copy*, but do not configure for display on any ticket screen.
 - Object type: Ticket
 - Field type: Text
 - Name: `service_copy`
 - Label: *Service Copy*
2. Create a generic agent job.
 - *Event Based Execution* section: Set the events `TicketCreate` and `TicketServiceUpdate` as triggers.
 - *Select Tickets* section: Set filters if the field should not be set globally (e. g. restricting this to specific queues).
 - *Execute Custom Module* section:
 - Module: `Kernel::System::GenericAgent::CopyTicketValueToDynamicField`
 - Param 1 key: `Ticket_Service`
 - Param 1 value: `service_copy`

Module parameters control which fields should be copied. The key determines the ticket attribute (service, queue, etc.) and the value determines the name of the destination dynamic field.

Note: `Ticket_` prefix is needed if `Services` is used as parameter key. It will be included on the values for setting new values on ticket and it is not possible to get a real value for it, then as consequence the value for this ticket field will be empty.

3. Mapping from dynamic field values can now be used to link service names to actual sender addresses.

24.2 Agent Interface

This package has no agent interface.

24.2.1 Mapped Values as Sender Address

When a ticket reply or forward is created or when an event based notification is sent, the configured dynamic field of the ticket is checked. If the dynamic field contains a value, a mapping configuration is checked. If there is a mapped value in *Sender Address Mapping*, this value will be used as sender address.

If the dynamic field is not set or no mapping value exists, the default queue-based sender address will be used.

This feature should be used if your tickets have dynamic fields that contain information which sender address should be used, but are not email addresses. For this purpose the dynamic field values can be mapped to email addresses.

Example Usage

A dynamic field *Customer location* is configured to be used as sender address base. The configured queue sender address is *support@company.com*.

In the mapping module the following is configured:

- England → *support.uk@company.com*
- France → *support.fr@company.com*
- Germany → *support.de@company.com*
- Scotland → *support.uk@company.com*

When an agent replies to a ticket, the *Customer location* field is looked up and depending on the location the sender address is determined.

- Dynamic field value: <empty field>, sender address: *support@company.com* (address from queue).
- Dynamic field value: England, sender address: *support.uk@company.com* (mapped value).
- Dynamic field value: Ireland, sender address: *support@company.com* (no mapping found, therefore address from queue).
- Dynamic field value: France, sender address: *support.fr@company.com* (mapped value).

Job Settings

★ Job name:

Validity:

▶ Automatic Execution (Multiple Tickets)

▼ Event Based Execution (Single Ticket)

Event Triggers:

TYPE	EVENT	DELETE
Ticket	TicketCreate	
Ticket	TicketServiceUpdate	

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:

To add a new event select the event object and event name.

▶ Select Tickets

▶ Update/Add Ticket Attributes

▶ Add Note

▶ Execute Ticket Commands

▼ 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:

Save Changes

or Cancel

24.2.2 Direct Values as Sender Address

When a ticket reply or forward is created or when an event based notification is sent, the configured dynamic field of the ticket is checked. If the dynamic field contains a value, this value will be used as sender address.

If the dynamic field is not set, the default queue-based sender address will be used.

This feature should be used if your tickets have dynamic fields that contain email addresses which should be used as sender address.

Note: With this feature dynamic field values are not checked in advance, therefore all values for the configured field need to be valid email addresses to prevent delivery errors.

Example Usage

A dynamic field *Customer location address* is configured to be used as sender address. The configured queue sender address is *support@company.com*. When an agent replies to a ticket, the *Customer location address* field is looked up and the sender address is determined.

- Dynamic field value: <empty field>, sender address: support@company.com (address from queue).
- Dynamic field value: France, sender address: France (dynamic field value is an invalid email address, the email will not send).
- Dynamic field value: support.uk@company.com, sender address: support.uk@company.com (dynamic field value).

24.3 External Interface

This package has no external interface.

Escalation Suspend

This feature add-on not only makes **OTRS** an even better tool for solving problems quickly and shortening response times, it also provides greater precision when it comes to time accounting. If you need to pause an escalation, this feature add-on will do the job perfectly. The feature add-on allows you to configure one or more statuses that will put one or more escalations (SLAs) automatically on hold. For example, if you are waiting for a third party answer or for a process step to be carried out, this feature add-on will give you more freedom by allowing you to stop the escalation countdown. Removing the state will then reactivate the escalation(s) automatically, and the time left to process the ticket will be displayed once again.

This configurable status is basically a pause function for escalations. The time that passes when the feature add-on is active is not counted in the ticket's resolution time. This provides greater time accounting precision.

Benefits

- Minimizes unnecessary escalations.
- Enables precise time accounting.
- Provides greater flexibility through its configurable status.

Target Groups

- IT service management
- Internal IT
- Customer service

Available in Service Package

- GOLD

Package Name in OTRS Package Manager

- OTRSEscalationSuspend

25.1 Administrator Interface

This package has no administrator interface, but includes the functionality to stop or suspend escalations. If a ticket is assigned to one of several configurable states, the escalation is stopped (disabled).

25.1.1 Configuration

To configure the states where the escalation is stopped:

1. Go to *System Configuration* screen.
2. Select *OTRSEscalationSuspend* in the *Navigation* widget.
3. Navigate to *Core* → *Ticket* in the navigation tree.
4. Add new entries to setting `EscalationSuspendStates`.
5. If suspend escalation is needed for already escalated tickets, enable the `SuspendEscalatedTickets` setting.

Note: Invalid states will not be considered as suspend states for the calculation.

When changing the state of the ticket to normal, the escalation will continue. In this case, it starts with the date when the status was changed, displaying the remaining time.

Therefore, the entire period when the ticket was not on a normal state is not counted for solution time. However, only periods when the ticket was in suspended states before first response is not counted for first response time.

25.1.2 Console Command

This package contains a console command `Maint::Ticket::RebuildEscalationIndexOnline` handled by the OTRS daemon used for resetting the escalation times to the point it was suspended. This script is the one in charge to reset the escalation time to the point it was suspended.

To check if the script is running, execute the following command and find the `RebuildEscalationIndexOnline` task in the *Recurrent cron tasks* section.

```
otrs> /opt/otrs/bin/otrs.Console.pl Maint::Daemon::Summary
```

25.2 Agent Interface

This package has no dedicated agent interface.

Note: For this example escalations should be configured for at least one queue.

Example usage:

1. Go to one of the new ticket screens, and create a new ticket for one of the queues configured with escalation (for this example *Escalation - first response time* is OK).
2. Open the ticket detail view for the newly created ticket. Notice that the escalation is running.

- Once the escalation is running, change the ticket state to one that has been configured in `EscalationSuspendStates` setting.
- Go back to the ticket detail view for the suspended escalation ticket and verify that the escalation is not running.
- Change the ticket state to one different from those configured on `EscalationSuspendStates` setting. The escalation continues.

There is a new filter *Suspended Escalations* for the ticket lists. Select the *Remove tickets with suspended escalations* checkbox to remove all tickets from the list that are in an escalation suspending state.

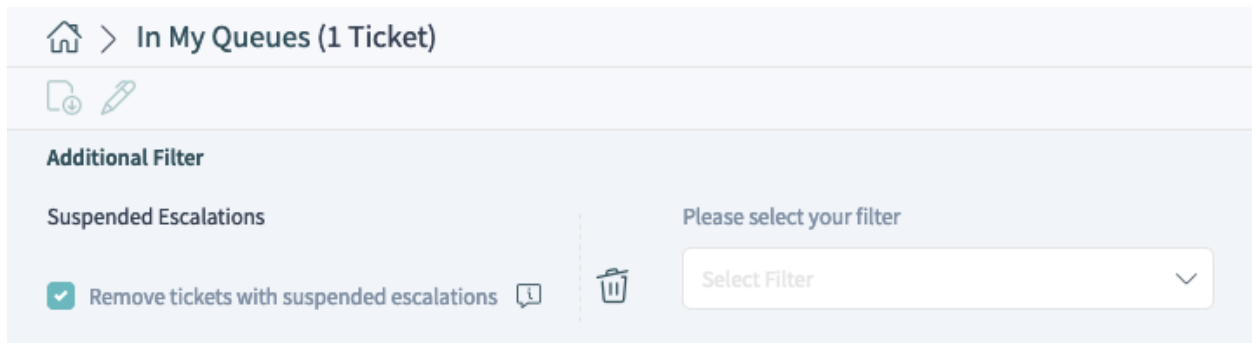


Fig. 1: Suspended Escalations Filter in Ticket List

Several ticket list screens also provide a default filter preset called *Escalation is not suspended*, which applies the aforementioned filter.

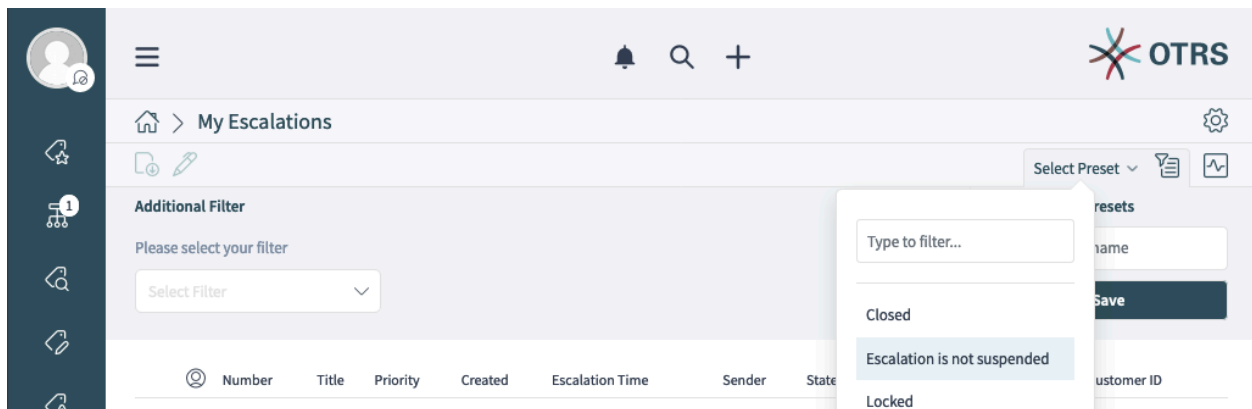


Fig. 2: Filter Preset in *My Escalations* Organizer Item

25.3 External Interface

This package has no external interface.

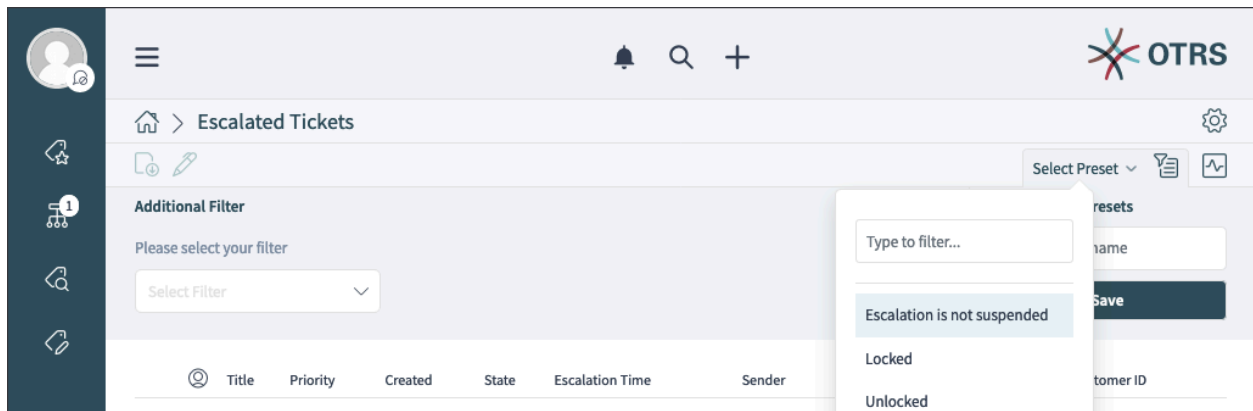


Fig. 3: Filter Preset in *Show Escalations / Escalated Tickets* Screen

Extended Ticket Stats

Personalize your statistics using a wide range of configuration options: First, specify the name, the assignment of rights and the format of your statistic. Then, you can select the desired attributes. These include:

- Ticket number, type and status
- Queue
- Ticket creation date
- Ticket closing date
- Rate of resolution
- Rate of resolution, adjusted for account work calendars
- Join and leave times for queues

Each chosen attribute is displayed as distinct value series in the statistics. Afterwards, decide whether to sort the data in ascending or descending order and define its absolute or relative time frame.

Benefits

- Obtain detailed information regarding the duration of tickets in your **OTRS**.
- Find clues for optimizing your processes.
- Detect employee overload at an early stage and take timely measures.

Target Groups

- IT services
- Call centers
- Customer service manager
- Project management
- Senior management

Available in Service Package

- PLATINUM

Package Name in OTRS Package Manager

- OTRSExtendedTicketStats

26.1 Administrator Interface

This package has no administrator interface.

26.2 Agent Interface

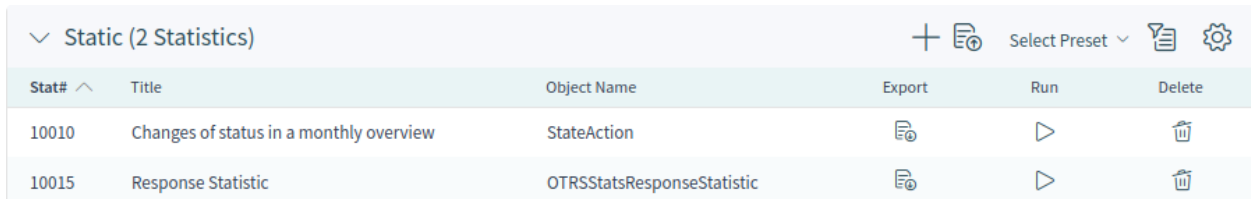
This package extends the statistic feature with adding new statistic modules for ticket analysis. The package also adds a new ready to use statistic.

26.2.1 Statistics and Reports

After installation of the package, a new statistic and two new list objects will be available for the statistics.

New Statistic

After installation of the package, a new static statistic will be available in the *Static* widget of the *Statistics and Reports* screen.



Static (2 Statistics)			+	+	Select Preset ▾	+	⚙️
Stat# ^	Title	Object Name	Export	Run	Delete		
10010	Changes of status in a monthly overview	StateAction	📄	▶	🗑️		
10015	Response Statistic	OTRSStatsResponseStatistic	📄	▶	🗑️		

Fig. 1: Static Widget

Response Statistic This static type statistic makes it possible to count how many times a response template has been used on a time frame. The result can be filtered by a time period and by queues.

To use the new statistic:

1. Search for *Response Statistic* in the *Static* widget.
2. Click on the *Run* icon in the column.
3. Select the time frame and the queues to include in the statistic.
4. Generate your statistic.

New List Objects

After installation of the package, two new list objects will be added to the system.

✓ Configurable parameters of static statistic

Start Day

 x ∨

Start Month

 x ∨

Start Year

 x ∨

End Day

 x ∨

End Month

 x ∨

End Year

 x ∨

Queues

 x ∨

Fig. 2: Response Statistic Settings

Statistics for Queue Resting Time

You are able to calculate the resting time of tickets for queues with this statistic module. The statistic considering also the close times of tickets. This accounted time will not added to the resting time or the resting time based on working period. The resting time based on working periods will be only displayed if you have configured calendars for your queues.

Available columns are:

- Number: numeration of the ticket list.
- Ticket#: ticket number.
- Ticket ID: internal ID of the ticket.
- Created: ticket creation date.
- Changed: last change date of the ticket.
- Closed: last close date of the ticket.
- Status: current state of the ticket.
- Type: current type of the ticket.
- Queue: current queue based on the history entries.
- Queue join time: date based on history entries (ticket created in this queue, moved into this queue while being open or reopened in this queue).
- Queue left time: date based on history entries (ticket closed in this queue or moved out of this queue while being open). Empty if ticket was still open and in this queue at end of reporting time-frame.
- Resting time (seconds): count of the duration when the ticket stayed in the queue, based on join and left time. Times in which a ticket was closed will not be counted.
- Resting time (HH:MM:SS): count of the duration when the ticket stayed in the queue, formatted as hours, minutes and seconds.
- Resting time (Working hours, seconds): count of the duration when the ticket stayed in the queue based on the queue configured working calendars in the system configuration (this field will be empty if no calendar is configured).
- Resting time (Working hours, HH:MM:SS): formatted count of the duration when the ticket stayed in the queue based on the queue configured working calendars in the system configuration (this field will be empty if no calendar is configured).
- Criticality: current criticality of the ticket.
- Impact: current impact of the ticket.

To use the new list object:

1. Create a new *List* statistic.
2. Choose the object `OTRSStatsQueueRestingTime`.
3. Configure the rest of the configuration options according to your wishes.
4. Run the statistic with output print or CSV.
5. All tickets matching the restrictions will be shown. Additional tickets will be shown, if they were created before the beginning of the reporting and were still open during the reporting time frame.

Statistics for Queue Working Time

This statistic is a modified ticket list to calculate the working time per queue. This modified ticket list statistic contains 3 new columns for the X-axis. In general the time for the 3 new columns is calculated on the time where ticket was not in a closed, removed or merged state type.

All queues in the system have the *Time*, *Working Time* and *Effective Time* column for this statistic object.

Example queue *Misc*:

- *Queue Misc Time* contains the time the agent have worked on the ticket.
- *Queue Misc Working Time* contains the time the agent have worked on the ticket based on the working calendars of the queue *Misc*.
- *Queue Misc Effective Time* contains the time the agent set as time units for the ticket while it was in the queue *Misc*. The values are displayed for each agent by first and last name in the same column (comma separated).

To use the new list object:

1. Create a new *List* statistic.
2. Choose the object `OTRSStatsQueueWorkingTime`.
3. In the *X-axis* section, select the columns for the queue you like to display the working time or time.
4. Set all other restrictions and sort options you need for the statistic to complete the configuration.
5. Generate your statistic.

26.3 External Interface

This package has no external interface.

Hide/Show Dynamic Fields

A dynamic field enables the extension of the information stored in a ticket and the individual configuration of your **OTRS**. However, dynamic fields can also cause confusion when they appear in the wrong context.

A typical example of an internal IT department process is the ordering of hardware or software in **OTRS**. Dynamic fields help structure this order process through drop-down lists. For example, after choosing the hardware to be ordered, the feature add-on ensures that only appropriate choices will appear in the next step, e.g. screen, keyboard and printer. The dynamic fields defined for software, like word processing, image processing or spreadsheet software, will not be displayed.

Benefits

- Clarity and transparency for your users.
- Flexible configuration of ticket information in the agent and external interface.
- Faster and more accurate entering of information, saving time for your employees and customers.

Target Groups

- IT services
- Complaint management
- E-commerce
- Supply-chain management
- IT support
- Sales
- Marketing
- Procurement

Available in Service Package

- GOLD

Package Name in OTRS Package Manager

- OTRSHideShowDynamicFields

27.1 Administrator Interface

This package has no administrator interface, but the function can be used in agent and external interface. The following screens can be configured.

Agent interface:

- New Email Ticket
- New Phone Ticket
- New Process Ticket (incl. Process Activity Dialogs)
- New SMS Ticket
- Close Ticket
- Change Customer
- Send Email Outbound
- Change Free Fields
- Link Objects
- Merge Ticket
- Move Ticket
- Add Note
- Change Owner
- Set Pending Time
- Add Phone Call Inbound
- Add Phone Call Outbound
- Change Priority
- Change Responsible
- Send SMS Outbound
- Forward via Email
- Redirect via Email
- Reply via Email
- Reply to All via Email
- Reply via Note
- Reply via SMS

External interface:

- Create New Ticket
- Create New Process Ticket (incl. Process Activity Dialogs)
- Ticket Reply (within Ticket Detail View)

27.1.1 Example Usage

Goals:

- If brand *VW* is selected, all dynamic fields should be hidden and just *VW Model* is displayed.
- If *VW* model *Up* is selected, all dynamic fields should be displayed except for the fields *Peugeot Model* and *Peugeot Production Facility*.

Create the following dynamic fields:

Object	Type	Name	Label	Possible values
Ticket	Dropdown	Brand	Brand	<ul style="list-style-type: none"> VW → VW Peugeot → Peugeot
Ticket	Dropdown	VWModel	VW Model	<ul style="list-style-type: none"> Up → Up Polo → Polo Golf → Golf T5 → T5
Ticket	Dropdown	VWProductionFacility	VW Production Facility	<ul style="list-style-type: none"> Barcelona → Barcelona Berlin → Berlin Bratislava → Bratislava
Ticket	Dropdown	PeugeotModel	Peugeot Model	<ul style="list-style-type: none"> 207 → 207 307 → 307
Ticket	Dropdown	PeugeotProductionFacility	Peugeot Production Facility	<ul style="list-style-type: none"> Poissy → Poissy Madrid → Madrid Trnava → Trnava
Ticket	Dropdown	Fuel	Fuel	<ul style="list-style-type: none"> Gasoline → Gasoline Diesel → Diesel Gas → Gas
Ticket	Multiselect	Accessories	Accessories	<ul style="list-style-type: none"> CDRadio → CD Radio GPS → GPS ProximitySensors → Proximity Sensors RearCamera → Rear Camera ClimateControl → Climate Control
194			Chapter 27. Hide/Show Dynamic Fields	
Ticket	Textarea	Remarks	Remarks	
Ticket	Date	RegistrationDate	Registration Date	
Ticket	Date	InvoiceDate	Invoice Date	

Note: All *Dropdown* and *Multiselect* dynamic fields should have the option *Add empty value* set to *Yes* in their configuration.

Add the dynamic fields to the *New Phone Ticket* screen via setting `Forms###AgentFrontend::TicketCreate::Phone::CreateProperties:`

```
- Label: Dynamic Fields
Collapsible: 1
Fields:
- Name: DynamicField_Brand
- Name: DynamicField_VWModel
- Name: DynamicField_VWProductionFacility
- Name: DynamicField_PeugeotModel
- Name: DynamicField_PeugeotProductionFacility
- Name: DynamicField_Fuel
- Name: DynamicField_Accessories
- Name: DynamicField_Remarks
- Name: DynamicField_RegistrationDate
- Name: DynamicField_InvoiceDate
```

Import this ACL:

```
---
- ChangeBy: root@localhost
ChangeTime: 2019-07-22 11:44:25
Comment: ''
ConfigChange:
  PossibleNot:
    Form:
    - PeugeotModel
    - PeugeotProductionFacility
    - Accessories
    - Fuel
    - Remarks
    - RegistrationDate
    - InvoiceDate
  ConfigMatch:
    Properties:
    Ticket:
      DynamicField_Brand:
      - VW
CreateBy: root@localhost
CreateTime: 2019-07-22 11:40:43
Description: ''
ID: 1
Name: ACL-VW
StopAfterMatch: 0
ValidID: 1
```

Detailed explanation:

```
DynamicField_Brand:
- VW
```

The condition for this ACL rule. If brand *VW* is selected, the rule will come into action. The array contains the used possible values. These are keys found in your database inside the `dynamic_field` table in column `config`. In this example it is a dynamic field of type *Dropdown*.

Form:

This package introduces the `Form` key as new option in the *Possible*, *PossibleAdd* and *PossibleNot* ACL change sections. `Form` holds the configuration for the visibility of dynamic fields.

PossibleNot:

Form:

- PeugeotModel
- PeugeotProductionFacility
- Accessories
- Fuel
- Remarks
- RegistrationDate
- InvoiceDate

This section lists the dynamic fields that should not be visible. In this example the dynamic fields *VW Model* and *VW Production Facility* are visible. All other dynamic fields will be hidden.

Import this second ACL:

```

---
- ChangeBy: root@localhost
ChangeTime: 2019-07-22 12:06:24
Comment: ''
ConfigChange:
  Possible:
    Ticket:
      DynamicField_Accessories:
        - CDRadio
        - ClimateControl
      DynamicField_Fuel:
        - Gasoline
      DynamicField_VWProductionFacility:
        - Bratislava
    PossibleAdd:
      Form:
        - Accessories
        - Fuel
        - Remarks
        - RegistrationDate
        - InvoiceDate
    PossibleNot:
      Form:
        - PeugeotModel
        - PeugeotProductionFacility
  ConfigMatch:
    Properties:
      Ticket:
        DynamicField_Brand:
          - VW

```

(continues on next page)

(continued from previous page)

```

DynamicField_VWModel:
  - Up
CreateBy: root@localhost
CreateTime: 2019-07-22 11:47:02
Description: ''
ID: 2
Name: ACL-VW-Up
StopAfterMatch: 0
ValidID: 1

```

Detailed explanation:

```

DynamicField_Brand:
  - VW
DynamicField_VWModel:
  - Up

```

In this example two conditions should be met. Brand has to be *VW* and VW model has to be *Up* for this rule to come into action. It will be triggered only if an agent selects brand *VW* **and** VW model *Up*.

```

PossibleAdd:
Form:
  - Accessories
  - Fuel
  - Remarks
  - RegistrationDate
  - InvoiceDate

```

Here the dynamic fields *VW Model* and *VW Production Facility* were already visible and they remain, but *Accessories*, *Fuel*, *Remarks*, *Registration Date* and *Invoice Date* has to be re-added to the fields that are visible. This is done in the *PossibleAdd* section as the first ACL sets this fields as not shown and both ACLs works together. If this was done in the *Possible* section for example, the result will be that only these fields explicitly will be shown and *VW Model* and *VW Production Facility* will be hidden as they are not longer part of the (new) *Possible* section.

```

PossibleNot:
Form:
  - PeugeotModel
  - PeugeotProductionFacility

```

Just *Peugeot Model* and *Peugeot Production Facility* are invisible (in our example it does not make much sense to configure a Peugeot model if the user has a VW Up).

In addition to the visibility of dynamic fields there is the possibility to show just some of the possible values of a dynamic field. Combined into ACL rules like in here, makes it easier to handle big multiselects.

```

Possible:
Ticket:
DynamicField_Accessories:
  - CDRadio
  - ClimateControl
DynamicField_Fuel:
  - Gasoline

```

(continues on next page)

(continued from previous page)

DynamicField_VWProductionFacility:

- Bratislava

In our example a VW Up can have just CD Radio and Climate Control as extra accessories, just Gasoline as fuel and can be produced just in Bratislava.

If we would have an ACL rule for Peugeot 207 for example, there may be other extras, fuel options and production locations selectable.

Note: If you are showing dynamic fields using the *Possible* option based on a `DynamicField_NameX` value, is normally desirable to include the dynamic field that triggers the ACL to be part of the fields to be displayed in the *Possible* or *PossibleAdd* sections (if apply). Otherwise if *Possible* or *PossibleAdd* contains other fields and not the trigger, the latest will not be shown after the value is selected.

Note: The mandatory status of the fields can not be changed using this method.

27.2 Agent Interface

This package has no agent interface.

27.3 External Interface

This package has no external interface.

Link Object Connector

The main feature of this package is to extend the OTRS generic interface that enables OTRS to accept requests from external systems to interact with the link object.

Benefits

- Accept requests from external systems to interact with the link object.

Target Groups

- Development
- IT service

Available in Service Package

- GOLD

Package Name in OTRS Package Manager

- OTRSLinkObjectConnector

28.1 Administrator Interface

This package has no administrator interface, but extends the web service with new operations.

See also:

In order to use this feature, a new web service has to be configured in the [Web Services](#) screen of the administrator interface.

28.1.1 Processes & Automation

The functionality needs to be configured in the *Web Services* module of the administrator interface. The package contains example configurations for both SOAP and REST methods.

Web Services

The functionality needs to be configured in the *Web Services* module of the administrator interface. The package contains example configurations for both SOAP and REST methods.

To import the web service configuration:

1. Go to the *Web Service Management* screen in the administrator interface.
2. Click on the *Add Web Service* button in the left sidebar.
3. Click on the *Import Web Service* button in the left sidebar.
4. Select the file `scripts/GenericLinkConnectorSOAP.yml` or `scripts/GenericLinkConnectorREST.yml` within the OTRS home folder.
5. Click on the *Import* button.

These files contain the complete information to setup the web service. The web service can be customized after the import process, if needed. The result of this process will be the same as the manual procedure explained below, but with some useful comments are added to the operations. The imported web services are ready and able to handle all the new operations provided in this package.

It is possible to setup the configuration manually, if you would not like to use the builtin configuration examples.

Note: The following steps are needed only, when the web services were not imported as explained above.

The following steps are required to provide a full working configuration:

1. Go to the *Web Service Management* screen in the administrator interface.
2. Click on the *Add Web Service* button in the left sidebar.
3. Fill the form with the following data:
 - Name: LinkObjectConnector
 - Debug threshold: Debug
 - Network transport: HTTP::SOAP
4. Click on the *Save* button to enable the *Add Operation* selection and the *Configure* button for network transport.
5. Click on the network transport *Configure* button to show the transport options.
6. Use the following parameters:
 - Namespace: <https://otrs.org/LinkObjectConnector/>
 - Maximum message length: 100000
7. Click on the *Save* button. Wait until the page refreshes and then click on the *Go back to web service* button to continue.
8. The *Add Operation* selection provides seven new operations for link:

```
Link::LinkAdd
Link::LinkDelete
Link::LinkDeleteAll
Link::LinkList
Link::PossibleLinkList
```

(continues on next page)

(continued from previous page)

```
Link::PossibleObjectsLst
Link::PossibleTypesList
```

9. Add the `Link::LinkAdd` operation and set the name `LinkAdd` in the next screen. It does not need any further configuration.
10. Click on the *Save* button. Wait until the page refreshes and then click on the *Go back to web service* button to continue.
11. Repeat step 9 and step 10 for the other six `Link::` operations.
12. After adding all operations, click on the *Save and Finish* button from the *Edit Web Service: LinkObjectConnector* screen.

Now you are on the *Web Service Management* overview screen and the new web service is ready and able to handle all the new operations provided in this package.

Note: When the new web service is fully tested, the debug threshold could be changed to *Notice*. This will reduce the amount of data that will be kept in the database and only notice and error messages will be logged.

The main feature of this package is to extend the OTRS generic interface that enables OTRS to accept requests from external systems to interact with the link object.

Each operation requires a valid agent user login name and a password or a session ID. This session ID can be obtained using the `SessionCreate` operation from session connector that is available by default in OTRS.

This package includes the following generic interface operations.

Link::LinkAdd This operation is used to create a link between two objects.

Link::LinkDelete This operation is used to remove a link between two objects.

Link::LinkDeleteAll This operation is used to remove all link of an object.

Link::LinkList This operation shows all links of an object, optionally restricted by another object, link type and link direction.

Link::PossibleLinkList This operation shows all possible link types between objects that are registered in the OTRS system.

Link::PossibleObjectsList This operation shows all objects that can be used for linking.

Link::PossibleTypesList This operation shows all possible link types between two given objects.

Due to the nature of the generic interface and the operations included in this package an external software is needed to send the requests to the OTRS system.

For testing we recommend the use of:

- OTRS Perl SOAP Requester script: some of the variables in this script such as `URL`, `NameSpace` and `Operation` will need to be changed to match the current web service, the operation to be executed and the data to be sent.
- [SoapUI](#) by SMARTBEAR: this is an open source software designed to test web services using SOAP messages.

28.1.2 Usage

The link object feature provides functionality to link items (e.g. tickets, changes) from objects (e.g. Ticket, ITSMChangeManagement) in order to provide a logical connection.

To make easier to read and understand these requests here are some SOAP request templates showing the parameters for each operation.

Link: :LinkAdd Request Template

```
<RootElement>
  <UserLogin?></UserLogin>
  <Password?></Password>
  <SessionID?></SessionID>
  <SourceObject?></SourceObject>
  <SourceKey?></SourceKey>
  <TargetObject?></TargetObject>
  <TargetKey?></TargetKey>
  <Type?></Type>
  <State?></State>
</RootElement>
```

UserLogin and Password Used for authentication. This is optional, but either a user login with a password or a session ID have to be provided.

SessionID Used for authentication. This is optional, but either a user login with a password or a session ID have to be provided.

SourceObject Object where the link originates.

SourceKey ID of the originating object.

TargetObject Object where the link points to.

TargetKey ID of the target object.

Type Type of the link.

State State of the link, usually Valid.

Link: :LinkDelete Request Template

```
<RootElement>
  <UserLogin?></UserLogin>
  <Password?></Password>
  <SessionID?></SessionID>
  <Object1?></Object1>
  <Key1?></Key1>
  <Object2?></Object2>
  <Key2?></Key2>
  <Type?></Type>
</RootElement>
```

UserLogin and Password Used for authentication. This is optional, but either a user login with a password or a session ID have to be provided.

SessionID Used for authentication. This is optional, but either a user login with a password or a session ID have to be provided.

Object1 Object where the link originates.

Key1 ID of the originating object.

Object2 Object where the link points to.

Key2 ID of the target object.

Type Type of the link.

Link: :LinkDeleteAll Request Template

```
<RootElement>
  <UserLogin>?</UserLogin>
  <Password>?</Password>
  <SessionID>?</SessionID>
  <Object>?</Object>
  <Key>?</Key>
</RootElement>
```

UserLogin and Password Used for authentication. This is optional, but either a user login with a password or a session ID have to be provided.

SessionID Used for authentication. This is optional, but either a user login with a password or a session ID have to be provided.

Object Object where the links originate.

Key ID of the originating object.

Link: :LinkList Request Template

```
<RootElement>
  <UserLogin>?</UserLogin>
  <Password>?</Password>
  <SessionID>?</SessionID>
  <Object>?</Object>
  <Key>?</Key>
  <Object2>?</Object2>
  <State>?</State>
  <Type>?</Type>
  <Direction>?</Direction>
</RootElement>
```

UserLogin and Password Used for authentication. This is optional, but either a user login with a password or a session ID have to be provided.

SessionID Used for authentication. This is optional, but either a user login with a password or a session ID have to be provided.

Object Object where the links originate.

Key ID of the originating object.

Object2 Optional restriction of the object where the links point to.

State State of the links, usually `Valid`.

Type Optional restriction of the link type.

Direction Optional restriction of the link direction (`Source` or `Target`).

Link::PossibleLinkList Request Template

```
<RootElement>
  <UserLogin>?</UserLogin>
  <Password>?</Password>
  <SessionID>?</SessionID>
</RootElement>
```

UserLogin and Password Used for authentication. This is optional, but either a user login with a password or a session ID have to be provided.

SessionID Used for authentication. This is optional, but either a user login with a password or a session ID have to be provided.

Link::PossibleObjectsList Request Template

```
<RootElement>
  <UserLogin>?</UserLogin>
  <Password>?</Password>
  <SessionID>?</SessionID>
  <Object>?</Object>
</RootElement>
```

UserLogin and Password Used for authentication. This is optional, but either a user login with a password or a session ID have to be provided.

SessionID Used for authentication. This is optional, but either a user login with a password or a session ID have to be provided.

Object Object where the links originate.

Link::PossibleTypesList Request Template

```
<RootElement>
  <UserLogin>?</UserLogin>
  <Password>?</Password>
  <SessionID>?</SessionID>
  <Object1>?</Object1>
  <Object2>?</Object2>
</RootElement>
```

UserLogin and Password Used for authentication. This is optional, but either a user login with a password or a session ID have to be provided.

SessionID Used for authentication. This is optional, but either a user login with a password or a session ID have to be provided.

Object1 Object where the link originates.

Object2 Object where the link points to.

28.1.3 Examples

The following examples use all objects and types that are available in OTRS using the default configuration. If you have changed the link default configurations, the examples might have to be modified.

Create New Link

This example will show how to create a link, using the `Link::LinkAdd` generic interface operation.

1. Send the following SOAP request to the OTRS system.

```
<RootElement>
  <UserLogin>johnsmith</UserLogin>
  <Password>Secret123</Password>
  <SourceObject>Ticket</SourceObject>
  <SourceKey>1</SourceKey>
  <TargetObject>Ticket</TargetObject>
  <TargetKey>2</TargetKey>
  <Type>Normal</Type>
  <State>Valid</State>
</RootElement>
```

2. The server response should be something similar to this one.

```
<namespace1:RootElementResponse>
  <Success>1</Success>
</namespace1:RootElementResponse>
```

Remove Link

This example will show how to remove the previously created link, using the `Link::LinkDelete` generic interface operation.

1. Send the following SOAP request to the OTRS system.

```
<RootElement>
  <UserLogin>johnsmith</UserLogin>
  <Password>Secret123</Password>
  <Object1>Ticket</Object1>
  <Key1>1</Key1>
  <Object2>Ticket</Object2>
  <Key2>2</Key2>
  <Type>Normal</Type>
</RootElement>
```

2. The server response should be something similar to this one.

```
<namespace1:RootElementResponse>
  <Success>1</Success>
</namespace1:RootElementResponse>
```

Remove All Links

This example will show how to alternatively remove the previously created link (and all other links of this ticket), using the `Link::LinkDeleteAll` generic interface operation.

1. Send the following SOAP request to the OTRS system.

```
<RootElement>
  <UserLogin>johnsmith</UserLogin>
  <Password>Secret123</Password>
  <Object>Ticket</Object>
  <Key>1</Key>
</RootElement>
```

2. The server response should be something similar to this one.

```
<namespace1:RootElementResponse>
  <Success>1</Success>
</namespace1:RootElementResponse>
```

Show All Links

This example will show all links of an object, using the `Link::LinkList` generic interface operation.

1. Send the following SOAP request to the OTRS system.

```
<RootElement>
  <UserLogin>johnsmith</UserLogin>
  <Password>Secret123</Password>
  <Object>Ticket</Object>
  <Key>1</Key>
  <State>Valid</State>
</RootElement>
```

2. The server response should be something similar to this one.

```
<namespace1:RootElementResponse>
  <RootElement>
    <Direction>Source</Direction>
    <Key>2</Key>
    <Object>Ticket</Object>
    <Type>Normal</Type>
  </RootElement>
  <RootElement>
    <Direction>Source</Direction>
    <Key>3</Key>
    <Object>Ticket</Object>
    <Type>ParentChild</Type>
  </RootElement>
  <RootElement>
    <Direction>Target</Direction>
    <Key>4</Key>
    <Object>Ticket</Object>
    <Type>ParentChild</Type>
```

(continues on next page)

(continued from previous page)

```

</RootElement>
</namespace1:RootElementResponse>

```

Show All Possible Links

This example will show all possible link types between objects, using the `Link::PossibleLinkList` generic interface operation.

1. Send the following SOAP request to the OTRS system.

```

<RootElement>
  <UserLogin>johnsmith</UserLogin>
  <Password>Secret123</Password>
</RootElement>

```

2. The server response should be something similar to this one.

```

<namespace1:RootElementResponse>
  <RootElement>
    <Object1>Ticket</Object1>
    <Object2>Ticket</Object2>
    <Type>Normal</Type>
  </RootElement>
  <RootElement>
    <Object1>Ticket</Object1>
    <Object2>Ticket</Object2>
    <Type>ParentChild</Type>
  </RootElement>
</namespace1:RootElementResponse>

```

Show All Possible Objects

This example will show all objects that can be linked to a given object, using the `Link::PossibleObjectsList` generic interface operation.

1. Send the following SOAP request to the OTRS system.

```

<RootElement>
  <UserLogin>johnsmith</UserLogin>
  <Password>Secret123</Password>
  <Object>Ticket</Object>
</RootElement>

```

2. The server response should be something similar to this one.

```

<namespace1:RootElementResponse>
  <PossibleObject>Ticket</PossibleObject>
</namespace1:RootElementResponse>

```

Show All Possible Types

This example will show all link types between two given objects, using the `Link::PossibleTypesList` generic interface operation.

1. Send the following SOAP request to the OTRS system.

```
<RootElement>
  <UserLogin>johnsmith</UserLogin>
  <Password>Secret123</Password>
  <Object1>Ticket</Object1>
  <Object2>Ticket</Object2>
</RootElement>
```

2. The server response should be something similar to this one.

```
<namespace1:RootElementResponse>
  <PossibleType>Normal</PossibleType>
  <PossibleType>ParentChild</PossibleType>
</namespace1:RootElementResponse>
```

28.2 Agent Interface

This package has no agent interface.

28.3 External Interface

This package has no external interface.

This feature add-on *ignores* out-of-office messages. As a rule, after processing a ticket, a final message is sent to a customer stating that the request has been completed. Then, the ticket is closed. If the message recipient has an out-of-office message turned on, this gets sent back to **OTRS** as a reply. This causes the closed ticket to be automatically re-opened. An employee must then manually close the ticket again. This can be very time-consuming, especially if you have frequent email traffic.

With the feature add-on, the system stops responding to out-of-office messages, which means tickets are not reopened when these messages are received. You can flexibly configure which out-of-office messages should be ignored.

Benefits

- Individual configuration that filters out-of-office messages.
- Closed tickets are no longer opened by out-of-office replies.

Target Groups

- Companies of all sizes with high ticket volumes
- Anyone using out-of-office frequently
- Companies with frequent email traffic
- IT service management
- Customer service
- Service providers

Available in Service Package

- TITANIUM

Package Name in OTRS Package Manager

- OTRSOutOfOffice

29.1 Administrator Interface

This package has no administrator interface.

29.1.1 Configuration

This package includes the functionality to create articles for a ticket with the `X-OTRS-Header` out of office without changing the status of the ticket. The module also prevents the locking of the ticket if mentioned header is present.

To configure this feature:

1. Go to *System Configuration* screen.
2. Select *OTRSOutOfOffice* in the *Navigation* widget.
3. Navigate to *Core* → *Email* → *PostMaster* in the navigation tree.
4. Activate the setting `OTRSOutOfOffice-Header`.
5. Select *OTRS* in the *Navigation* widget.
6. Navigate to *Core* → *Email* → *PostMaster* in the navigation tree.
7. Add the `X-OTRS-OutOfOffice` header to the end of the list in setting `PostmasterX-Header`.
8. Deploy the settings.
9. Go to the *PostMaster Filters* screen of the administrator interface and add a new postmaster filter.
10. Set email header `X-OTRS-Header` with value `1` of an incoming email based on whatever regular expression or text you want to use (e.g. Subject: *[Out of Office]*).
11. Click on the *Save* button.

29.2 Agent Interface

This package has no agent interface.

29.3 External Interface

This package has no external interface.

Process Management Article Email

This feature add-on makes it possible to add email addresses to process management articles, as well as send them via email. This guarantees a more direct communication straight from the process and therefore increases the transparency of the process communication.

Benefits

- More flexible usage of the OTRS process management.
- Direct communication with customers or other agents straight from the process.
- Greater transparency for customers or other agents regarding the process step their request is in.
- More interactivity with the customer or other agents during the process.
- More efficient design of your business processes.

Target Groups

- Process managers
- External IT
- Internal IT
- Customer service
- Facility management
- And many more

Available in Service Package

- GOLD

Package Name in OTRS Package Manager

- OTRSProcessManagementArticleEmail

30.1 Administrator Interface

This package has no administrator interface, but the feature add-on adds a new communication channel that is selectable for articles.

30.1.1 Processes & Automation

After installation of the package, it is possible to select *Email* as communication channel for sending an article to a customer user.

Process Management

This feature adds the ability to add email addresses to process management articles and sends the article to the inserted recipients.

To add an email communication channel to process management article:

1. Go to the *Process Management* screen.
2. Click on the *User Task Activity Dialogs* item in the *Available Process Elements* widget in the left sidebar.
3. Click on the pencil icon to edit the user task activity dialog.
4. Drag the *Article* field from the *Available Fields* pool and drop into the *Assigned Fields* pool or click on a pencil icon of an already added *Article* field.
5. Select *Email* in the *Communication Channel* field.

The screenshot shows a dialog box titled "Edit Field Details: Article" with a close button (X) in the top right corner. The dialog contains the following fields:

- Description (short): [Empty text input field]
- Description (long): [Empty text input field]
- Default value: [Greyed-out text input field]
- Communication Channel: [Dropdown menu with "Email" selected]
- Is visible for customer: [Unchecked checkbox]
- Time units: [Dropdown menu with "Do not show Field" selected]

At the bottom of the dialog are two buttons: "Save" and "Cancel".

Fig. 1: Edit Field Details Window

6. Save and deploy the process.
7. Go to the *Ticket Notifications* screen of the administrator interface.
8. Create a corresponding ticket notification for the event `ArticleCreate` so that the email is sent.

Warning: In the same activity dialog, you can use either the above-mentioned article field with email or a regular article field. Using both fields together will cause a process error.

Now go to the *Tickets* and start a new process.

30.2 Agent Interface

This package has no agent interface, but the feature add-on adds the regular email fields to *New Process Ticket* screen.

30.2.1 Tickets

After installation of the package, it is possible to send an article to a customer user when an agent starts a new process ticket.

New Process Ticket

To send an email to a customer user via a process ticket:

1. Click on the *Process Ticket* menu item in the *Action* menu or click on the *Start Processes* menu item in the main menu.
2. Select a process from the drop-down list.
3. Enter your customer information as already known from email tickets.

On submitting the activity dialog, the article gets sent to the configured customer user.

30.3 External Interface

This package has no external interface.

Ready2Adopt ITSM Processes

Use the following predefined processes in the IT service management disciplines of service design, service operation and service transition:

Service Design:

- Availability Management (proactive)
- Availability Management (reactive)
- Catalog Management
- Continuity Management
- Service Design Coordination (individual design)
- Service Level Management
- Supplier Management

Service Operation:

- Access Management
- Event Management
- Incident Management
- Problem Management
- Request Fulfillment Management
- Service Validation and Testing

Service Transition:

- Change Evaluation
- Knowledge Management
- Release and Deployment Management
- Service Asset and Configuration Management

- Service Transition Planning and Support

Benefits

- Continual service improvement (CSI) through pre-defined ITIL processes.
- Higher efficiency and quality through proven ITIL-inspired processes.

Target Groups

- Customer service
- Facility management
- IT
- IT service
- Security management
- Service providers

Available in Service Package

- GOLD

Package Name in OTRS Package Manager

- OTRSReady2AdoptITSMProcesses

Note: This feature add-on requires the *Configuration Management* feature.

31.1 Administrator Interface

This package has no administrator interface, but it adds the possibility to import Ready2Adopt processes.

31.1.1 Processes & Automation

After installation of the package, a list of several properly defined processes will be available in the *Process Management* screen.

Process Management

This package includes several examples of predefined processes that can help in some specific cases.

The following Ready2Adopt processes are available from the *Ready2Adopt Processes* widget:

- Service Design::Availability Management (proactive)
- Service Design::Availability Management (reactive)
- Service Design::Catalogue Management
- Service Design::Continuity Management
- Service Design::Service Design Coordination (individual design)
- Service Design::Service Level Management
- Service Design::Supplier Management

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

To import a Ready2Adopt process:

1. Go to the *Process Management* screen in the administrator interface.
2. Select a process from the *Ready2Adopt Processes* widget in the left sidebar.
3. Click on the *Import Ready2Adopt process* button.
4. Deploy all processes.

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

31.2 Agent Interface

This package has no agent interface.

31.3 External Interface

This package has no external interface.

Ready2Adopt Processes

Many processes in our everyday working life are standardized and recurring – and thus are ideal for using a process template.

With templates, mistakes can be avoided and tasks can be done faster. Improved quality and efficiency are the result!

The following process templates are available in **OTRS**:

- Time off requests
- Room reservations
- Office supply requests
- Travel expenses
- Complaints about materials

Benefits

- Quick execution of routine tasks.
- Error reduction due to predefined processes.
- Time saved on recurring tasks.
- Quality assurance through specified processes.
- Overview of the current status of an edit.
- Automation options.

Target Groups

- Complaint management
- Customer service
- Facility management
- Human resources

Available in Service Package

- SILVER

Package Name in OTRS Package Manager

- OTRSReady2AdoptProcesses

32.1 Administrator Interface

This package has no administrator interface, but it adds the possibility to import Ready2Adopt processes.

32.1.1 Processes & Automation

After installation of the package, a list of several properly defined processes will be available in the *Process Management* screen.

Process Management

This package includes several examples of predefined processes that can help in some specific cases.

The following Ready2Adopt processes are available from the *Ready2Adopt Processes* widget:

- Conference Room Reservation
- Office Materials Management
- Order Request Management
- Request for Leave Management
- Start RMA
- Travel expense

To import a Ready2Adopt process:

1. Go to the *Process Management* screen in the administrator interface.
2. Select a process from the *Ready2Adopt Processes* widget in the left sidebar.
3. Click on the *Import Ready2Adopt process* button.
4. Deploy all processes.

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

32.2 Agent Interface

This package has no agent interface.

32.3 External Interface

This package has no external interface.

Ready2Adopt Web Services

Not only do you use a single instance of **OTRS**, but you may also use additional **OTRS** installations and other tools. This can make accessing data inconvenient. Luckily, the following web services provide a workaround:

- Jira connector
- Bugzilla connector
- OTRS-OTRS connector

Benefits

- All data is centrally accessible in **OTRS**.
- Execute arbitrary complex mappings through the XSLT mapping module.

Target Groups

- Customer service
- Development
- IT
- IT service
- Security management
- Service providers

Available in Service Package

- GOLD

Package Name in OTRS Package Manager

- OTRSReady2AdoptWebServices

Note: This feature add-on requires the *Ticket Invoker* feature.

33.1 Administrator Interface

This package has no administrator interface, but it adds the possibility to import Ready2Adopt web services.

33.1.1 Processes & Automation

After installation of the package, a list of several properly defined web services will be available in the *Web Service Management* screen.

Web Services

This package includes several examples of predefined web services created by experts.

The following Ready2Adopt web services are available from the *Ready2Adopt Web Services* widget:

- `BugzillaConnector` to create or update bugs on a remote Bugzilla server.
- `JIRAConnector` to create or update issues on a remote JIRA server.
- `OTRSConnector` to create or update tickets on a remote OTRS server.

To import a Ready2Adopt web service:

1. Go to the *Web Service Management* screen in the administrator interface.
2. Select a web service from the *Ready2Adopt Web Services* widget in the left sidebar.
3. Click on the *Import Ready2Adopt web service* button.

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

33.2 Agent Interface

This package has no agent interface.

33.3 External Interface

This package has no external interface.

Restore Pending Information

With this feature add-on, your agents can save time dealing with recurring questions or follow-up messages from a customer. When the state *pending* is selected again, certain fields will be completed automatically.

After your agent has answered a question from a customer user and the customer user wants to test the recommended solution, your agent can set the ticket status to *pending* and specify a reminder time, a subject and a text for internal documentation. If the customer user sends another question before the reminder time runs out, your agent can answer and select the *pending* status again without having to re-enter the reminder time, title and text, as this information is now added automatically. If the reminder time expires, the default values are used.

Benefits

- Fewer clicks and less work by using the memory function.
- Saves time managing recurring requests from a customer.

Target Groups

- IT support
- IT services
- Facility management
- Call centers
- Customer service

Available in Service Package

- SILVER

Package Name in OTRS Package Manager

- OTRSRestorePendingInformation

34.1 Administrator Interface

This package has no administrator interface.

34.2 Agent Interface

After installation of the package, the functionality is automatically integrated in the *Set Pending Time* action. If there was an unexpired pending time set before, the functionality pre-fills the fields with the restored information from the last pending reminder set by the agent. This includes the following fields, which are pre-filled in the *Set Pending Time* action:

- State
- Pending Time
- Subject
- Body (if the text comes from a note article created by the *Set Pending Time* action)

Note: The functionality works only, when the pending time is not expired. When the pending time is expired the standard functionality sets the usual default values.

34.2.1 Use Case Example

1. The agent sets a *pending* state with a pending time and other information like subject and body text in the ticket.
2. The pending time is stored now in the property card of the *Properties* widget in the ticket detail view.
3. Before the pending time is expired, an incoming customer communication resets the pending time (the customer answers with “thanks”).
4. With the functionality of this package the agent is able to restore the pending time within the *Set Pending Time* action easily.
5. When the agent opens the *Set Pending Time* action, the fields are pre-filled with the information from the last entry which set a pending time.

Note: There are several options to set a pending time, but the pre-fill functionality is only integrated in the *Set Pending Time* action.

Note: The text field is only pre-filled from note articles, since this is the type of article the *Set Pending Time* action creates.

34.3 External Interface

This package has no external interface.

Restrict Customer Data View

Service organizations that go global or have a wide range of products have to outsource their customer service to distribution partners or call centers. When that happens, the security of customer data is very important and only the appropriate third party providing the service should have access to customer information.

This feature add-on makes this defined access possible with the assignment of customer IDs to partner IDs. It enhances and specifies the customer information center of the standard version and whose customer data can be accessed by all agents. When customer data is retrieved from an LDAP server, the feature add-on checks for a partner ID and which customer IDs are assigned to it. Therefore, agents of a partner ID can only see customer data of the assigned customer ID. If there is no customer ID assigned to a partner ID, the agents are not allowed to see any customer data. If no partner ID has been created, all agents can see all customer data.

The first configuration steps after the installation are the configuration of the LDAP server connection and the mapping of partner IDs to customer IDs with the help of a graphical interface. The administration of those connections can also be managed in a separate dashboard.

Benefits

- Specific allocation of access permissions allows protection of your customer data.
- Simplifies the collaboration with subsidiaries, subcontractors and partners.

Target Groups

- International companies
- Call centers
- Companies offering a wide range of products

Available in Service Package

- GOLD

Package Name in OTRS Package Manager

- OTRSRestrictCustomerDataView

35.1 Administrator Interface

This package offers the possibility to restrict the access of certain agents to data of certain customers following a mapping table.

35.1.1 Users, Groups & Roles

After installation of the package, a new module *Partner ID Customer ID* appears in the *Users, Groups & Roles* group in the administrator interface. Here you can define which partners should be assigned to customers.

PartnerID CustomerID

Use this screen to manage mappings between partners and customers. The management screen is available in the *PartnerID CustomerID* module of the *Users, Groups & Roles* group.

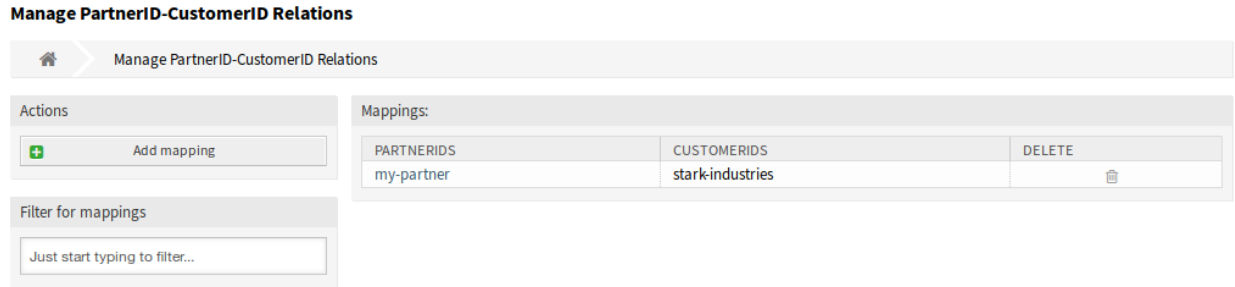


Fig. 1: Manage PartnerID-CustomerID Relations Screen

Manage PartnerID CustomerID Relations

Note: It is necessary to setup the LDAP server configuration in the system configuration.

To setup the LDAP server configuration:

1. Go to *System Configuration* screen.
2. Select *OTRSRestrictCustomerDataView* in the *Navigation* widget.
3. Navigate to *Core* → *LDAP* in the navigation tree.
4. Set the parameters for the LDAP connection.

If the LDAP server can be reached, the `customer_id` relations to the `partner_id` which is pulled from the LDAP server, will be stored in the database tables which are configured in the `PartnerIDMapping` section within the system configuration.

To see the `PartnerIDMapping` settings:

1. Go to *System Configuration* screen.
2. Select *OTRSRestrictCustomerDataView* in the *Navigation* widget.

3. Navigate to *Core* → *PartnerIDMapping* in the navigation tree.
4. See the settings.

Following the relations stored in the mapping tables, access to data of certain customers is restricted for agents which have a partner ID and relations in the database. Those agents can only see customers which have customer IDs that are allowed in the database.

If a partner ID of an agent is stored in the database, but no related customer IDs are assigned, the agent won't be able to see any customer information. Agents with no given partner ID will still be able to see every customer data.

If the LDAP server configuration and the partner ID mapping is properly set, you can proceed with the management screen.

To create a new mapping:

1. Click on the *Add mapping* button in the left sidebar.
2. Add a partner ID into the first field.
3. Add a customer into the second field. Just start typing, the field has auto-completion support. The added customer IDs will be displayed below the text field.
4. Click on the *Save* button.

The screenshot shows a form titled "Mappings:". It has two input fields: "PartnerID:" and "CustomerID:". Below the fields are two buttons: "Save" and "Cancel".

Fig. 2: Add Mapping Screen

To edit a mapping:

1. Click on a mapping in the list of mappings.
2. Add or remove customers.
3. Click on the *Save* or *Save and finish* button.

The screenshot shows a form titled "Mappings:". The "PartnerID:" field is filled with "my-partner". The "CustomerID:" field is empty, and a list of suggestions is shown below it, including "stark-industries". Below the fields are three buttons: "Save", "Save and finish", and "Cancel".

Fig. 3: Edit Mapping Screen

The *PartnerID* field is read-only. If you would like to edit the partner ID, you can delete the mapping and create a new one with the other partner ID.

To delete a mapping:

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

Mappings:

PARTNERIDS	CUSTOMERIDS	DELETE
my-partner	stark-industries	

Fig. 4: Delete Mapping Screen

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

Mapping Settings

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

PartnerID * The ID of a partner as stored in the LDAP.

CustomerID * The internal ID of a customer as stored in the OTRS database.

35.2 Agent Interface

This package has no agent interface.

35.3 External Interface

This package has no external interface.

Service-based Queue Routing

For a customer, excellent service initially starts with contacting the right service staff member. Being a help desk software, **OTRS** makes this possible with a clever queue and permission concept. However, if a ticket is created in the external or in the agent interface, one has to choose the suitable queue for his/her request from all existing and shown queues – not an easy task for a customer or responsible agent. For example a call center agent may not know which queue structure the service company has for the task on which he is working. This means a long decision-making times with a possible wrong result and the loss of precious working time.

This feature add-on now makes it possible to assign queues to several services, so that if this service is selected, then only the appropriate queues are shown. For example the service *Support Mobile Devices* can be assigned to the queues *IT-2nd Level-Support Hardware* and *IT-2nd Level Support Mobile Applications* so the right selection can be made more easily. This module can also be configured so that all queues can reappear with just one click. Time-consuming moves to the right queues and overextended customer users are a thing of the past.

Benefits

- Simplified assignment of tickets in the correct queue without comprehensive knowledge about the queue structures.
- Saving time.
- Reduced workload.

Target Groups

- Call centers
- Customer service
- Service providers
- IT service management

Available in Service Package

- SILVER

Package Name in OTRS Package Manager

- OTRSServiceBasedQueueRouting

36.1 Administrator Interface

After installation of the package a new module will be available in the *Ticket Settings* group of the administrator interface.

36.1.1 Ticket Settings

After installation of the package a new module will be available in the *Ticket Settings* group of the administrator interface.

Services Queues

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

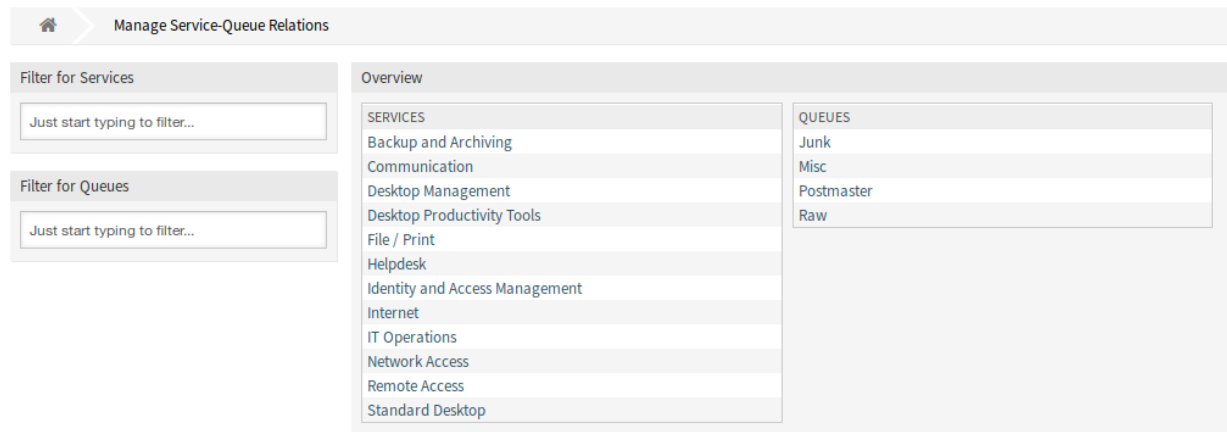


Fig. 1: Manage Service-Queue Relations Screen

Manage Services Queues Relations

To assign a service to a queue:

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

To assign a queue to a service:

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

Backup and Archiving

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

or Cancel

Fig. 2: Change Queue Relations for Service

Postmaster

SERVICE	<input type="checkbox"/> ACTIVE
Backup and Archiving	<input type="checkbox"/>
Communication	<input type="checkbox"/>
Desktop Management	<input type="checkbox"/>
Desktop Productivity Tools	<input type="checkbox"/>
File / Print	<input type="checkbox"/>
Helpdesk	<input type="checkbox"/>
Identity and Access Management	<input type="checkbox"/>
Internet	<input type="checkbox"/>
IT Operations	<input type="checkbox"/>
Network Access	<input type="checkbox"/>
Remote Access	<input type="checkbox"/>
Standard Desktop	<input type="checkbox"/>

or Cancel

Fig. 3: Change Service Relations for Queue

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

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

<p>Warning: Accessing a queue or a service provides no back link to the relations screen.</p>

36.2 Agent Interface

This package has no agent interface, but it extends the standard behavior of the queue selection method. The service based queue routing affects the ticket creation screens and the *Move Ticket* action. For this, the given service of the selected ticket will be used.

After selecting a service, there is a lookup for configured queues for this service. If there are configured queues for the selected service, only those will be available in the queue selection. If there are no queues configured for the selected service, all queues will be displayed.

It is possible to add a filter button next to the queue selection. After clicking this button all available queues get restored to the queue selection field and can be selected.

Configured ACLs affect the displayed queues as usual.

36.3 External Interface

This package has no external interface, but it extends the standard behavior of the queue selection method. The service based queue routing affects the ticket creation screens. For this, the given service of the selected ticket will be used.

After selecting a service, there is a lookup for configured queues for this service. If there are configured queues for the selected service, only those will be available in the queue selection. If there are no queues configured for the selected service, all queues will be displayed.

It is possible to add a filter button next to the queue selection. After clicking this button all available queues get restored to the queue selection field and can be selected.

Configured ACLs affect the displayed queues as usual.

Service Categories

If an agent or a customer creates a ticket and defines the ticket type, this feature add-on makes it easier to automatically display only the linked services from there on. The linking of ticket types and ticket services is carried out by an administrator with ticket configuration permissions, and it is possible to link a service to more than one ticket type, as well as a ticket type to more than one service. Ticket types and services can be additionally filtered and sorted in a drop-down menu.

Benefits

- Input help for agents and customers.
- Avoid process errors and incorrect assignments and the additional work they cause.
- Certain customer groups or request types can be processed more quickly.

Target Groups

- Internal and external IT service management
- Quality management
- Sales
- Facility management
- Customer satisfaction management
- Complaint management

Available in Service Package

- SILVER

Package Name in OTRS Package Manager

- OTRSServiceCategories

37.1 Administrator Interface

After installation of the package a new module will be available in the *Ticket Settings* group of the administrator interface.

37.1.1 Ticket Settings

After installation of the package a new module will be available in the *Ticket Settings* group of the administrator interface.

Types Services

Use this screen to add one or more types to one or more services. To use this function, at least one type and one service need to have been added to the system. The management screen is available in the *Types Services* module of the *Ticket Settings* group.

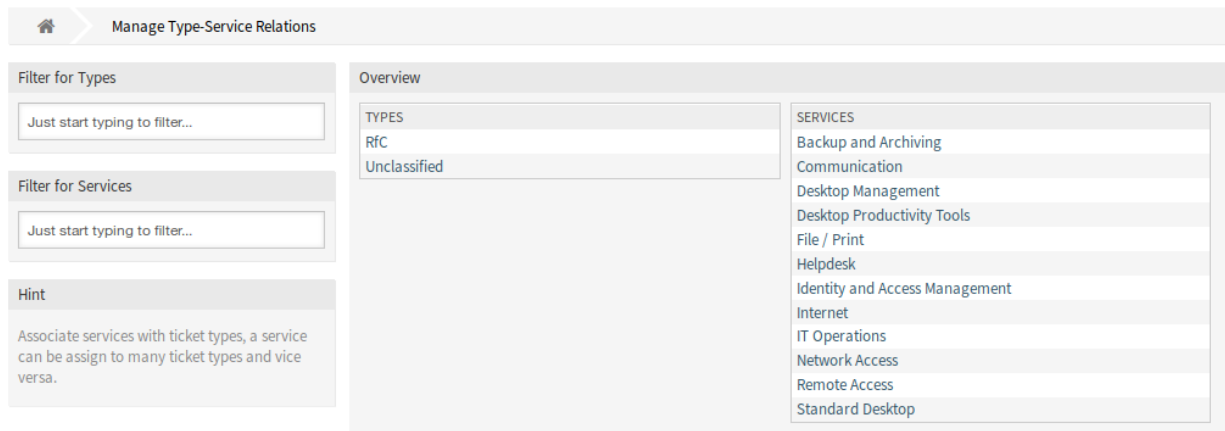


Fig. 1: Manage Type-Service Relations Screen

Manage Types Services Relations

To assign a type to a service:

1. Click on a type in the *Types* column.
2. Select the services you would like to add the type to.
3. Click on the *Save* or *Save and finish* button.

To assign a service to a type:

1. Click on a service in the *Services* column.
2. Select the types you would like to assign the service to.
3. Click on the *Save* or *Save and finish* button.

Change Service Relations for Type Unclassified

SERVICE	<input type="checkbox"/> ACTIVE
Backup and Archiving	<input type="checkbox"/>
Communication	<input type="checkbox"/>
Desktop Management	<input type="checkbox"/>
Desktop Productivity Tools	<input type="checkbox"/>
File / Print	<input type="checkbox"/>
Helpdesk	<input type="checkbox"/>
Identity and Access Management	<input type="checkbox"/>
Internet	<input type="checkbox"/>
IT Operations	<input type="checkbox"/>
Network Access	<input type="checkbox"/>
Remote Access	<input type="checkbox"/>
Standard Desktop	<input type="checkbox"/>

Save or **Save and finish** or Cancel

Fig. 2: Change Service Relations for Type

Change Type Relations for Service Backup and Archiving

TYPE	<input type="checkbox"/> ACTIVE
RfC	<input type="checkbox"/>
Unclassified	<input type="checkbox"/>

Save or **Save and finish** or Cancel

Fig. 3: Change Type Relations for Service

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

Multiple types or services can be assigned in both screens at the same time. Additionally clicking on a type or clicking on a service in the relations screen will open the *Edit Type* screen or the *Edit Service* screen accordingly.

Warning: Accessing a type or a service provides no back link to the relations screen.

37.2 Agent Interface

This package has no agent interface, but it extends the standard behavior of the service selection method.

37.2.1 Use Case Example

Service One and *Service Two* are assigned to type *Unclassified* in the *Types Services* screen. Other services are not assigned.

To filter the services on the ticket create screens:

1. Go to any ticket create screen, for example *New Phone Ticket*.
2. Choose *Unclassified* ticket type.
3. Fill in the other customer fields.
4. Check what services are available in the *Service* drop-down.

Note: *Customer Users Services* filter is applied before the one for *Types Services*.

37.3 External Interface

This package has no external interface.

State Pre-selection Response Templates

The use of response templates saves a lot of time – especially if an agent has to answer a lot of similar customer requests, such as requests for a quote or a manual.

In the standard version of **OTRS** you need to define the ticket state manually every time you use a response template. This is an additional task, which needlessly costs a lot of time and even is a source for mistakes, especially when you use response templates very often.

This feature add-on makes it possible to define a ticket state for every response template right from the start, so if the answer is sent, the ticket state is set automatically. This saves time and lowers the risk for mistakes.

Benefits

- Decreasing the risk of incorrect status inputs.
- Saving time.

Target Groups

- Customer service organizations having many partners or suppliers
- External IT service providers
- Call centers
- Sales departments and sales companies
- Advertising or communication agencies

Available in Service Package

- SILVER

Package Name in OTRS Package Manager

- OTRSStatePreselectionResponseTemplates

38.1 Administrator Interface


After installation of the package a new module will be available in the *Ticket Settings* group of the administrator interface.

38.1.1 Ticket Settings

This package adds a new module *State pre-selection for Templates* to the *Ticket Settings* group in the administration interface.

State Pre-selection for Templates

Use this screen to assign a ticket state to response templates. The ticket state will be pre-selected when answering a ticket via the response template. The management screen is available in the *State Pre-selection for Templates* module of the *Ticket Settings* group.



Manage ticket state pre-selections for response templates

Filter for Templates

Just start typing to filter...

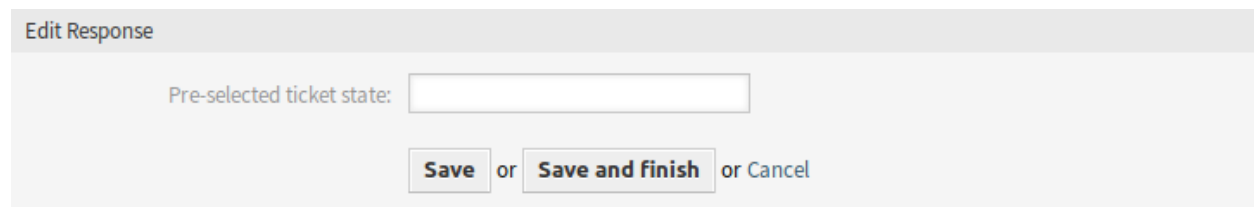
TEMPLATE NAME	PRE-SELECTED TICKET STATE
empty answer	-

Fig. 1: Ticket State Pre-selections For Response Templates Management Screen

Manage Responses

To assign a ticket state to a response template:

1. Click on a template in the list of templates.
2. Select a ticket state for the response template.
3. Click on the *Save* or *Save and finish* button.



Edit Response

Pre-selected ticket state:

Save or **Save and finish** or Cancel

Fig. 2: Edit Response Screen

Note: The configured ticket states must be allowed in ticket context as next state. If this should not be the case, no next state will be pre-selected when answering a ticket.

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

Response Settings

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

Pre-selected ticket state Select a ticket state to assign it to the template. This ticket state will be pre-selected on reply actions when answering a ticket using this response template.

38.2 Agent Interface

This package has no agent interface, but the package allows pre-selection of ticket states for response templates.

To pre-select a ticket state on reply:

1. Go to the ticket detail view and click on the *Reply via Email* action in the article menu.
2. Select a response template from the drop-down list.
3. The pre-selected ticket state should be set in the *State* field of the reply action.

See also:

The ticket state can be assigned in the [State Pre-selection for Templates](#) management screen.

38.3 External Interface

This package has no external interface.

System Configuration History

System configuration import and export are fully possible in the standard version of **OTRS**. An export always represents your complete configuration.

With this feature add-on, previous revisions can now be viewed and restored individually. This makes it easy to reset a setting, reducing errors and downtime. In addition, changes are auditable and revision-proof – the system registers which changes have been made by whom.

Benefits

- Error reduction by restoring configurations.
- Detailed change log for all settings.
- Accountability of changes.

Target Groups

- IT
- Security management

Available in Service Package

- SILVER

Package Name in OTRS Package Manager

- OTRSSystemConfigurationHistory

39.1 Administrator Interface

This package adds two new screens to see the history of the system configuration settings.

39.1.1 Administration

After installation of the package, new screens will be available for administrators to go through system configuration deployment and specific settings history, and make changes if needed.

System Configuration

The feature add-on adds new screens to the *Changes Deployment* screen and for the individual settings. These screens will display recent deployments in a descending order, along with their comments and name of the user who deployed the changes.

Deployment History

To see the deployment history:

1. Go to the *System Configuration* screen of the administrator interface.
2. Click on the *Deployment* button.
3. Click on the *History* button.

DATE	COMMENT	RESPONSIBLE	DETAILS	RESTORE	EXPORT
08/02/2019 09:52 (Europe/Budapest)	Package Install OTRSSystemConfigurationHistory 7.0.3	Admin OTRS	View Details		
08/02/2019 08:44 (Europe/Budapest)	Configuration Rebuild	Admin OTRS	View Details		
08/02/2019 08:42 (Europe/Budapest)	Configuration Rebuild	Admin OTRS	View Details		
08/01/2019 15:16 (Europe/Budapest)	Configuration Rebuild	Admin OTRS	View Details		
08/01/2019 15:15 (Europe/Budapest)	Configuration Rebuild	Admin OTRS	View Details		

Fig. 1: Deployment History Screen

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

Deployment Details
<p>Created: 07/29/2019 11:00 (Europe/Budapest) by Admin OTRS</p> <p>Comment: Enable responsible feature to add one more agent to tickets.</p>
<p>Ticket::Responsible Core::Ticket ⇄</p> <p><input checked="" type="checkbox"/> Enabled</p>

Fig. 2: Deployment Details Screen

It is possible to compare the old and the new settings side-by-side by clicking on the two arrows button.

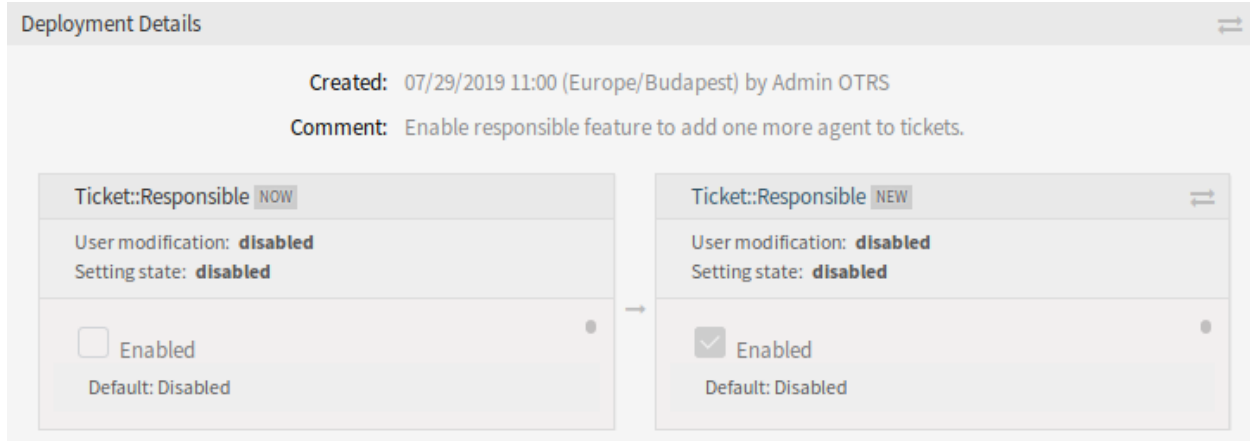


Fig. 3: Deployment Details Difference

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

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

Note: If several history entries are displayed in the history, use the filter box to find a particular log entry by just typing the name to filter.

Setting History

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

To see the setting history:

1. Go to the *System Configuration* screen of the administrator interface.
2. Search for a system configuration setting.
3. Open the setting for editing and click on its header to reveal the buttons.
4. Click on the *History* button.

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

39.2 Agent Interface

This package has no agent interface.

Change History of Ticket::Frontend::AgentTicketFreeText###DynamicField

Change History	Field	Value
On 07/03/2019 13:13 (Europe/Budapest) by Admin OTRS	Attachment	1 - Enabled
	Supervisor	1 - Enabled
	TicketCalendarEndTime	1 - Enabled
	TicketCalendarStartTime	1 - Enabled
On 06/17/2019 13:28 (Europe/Budapest) by Admin OTRS	Supervisor	1 - Enabled
	TicketCalendarEndTime	1 - Enabled
	TicketCalendarStartTime	1 - Enabled
On 03/22/2019 10:02 (Europe/Budapest) by Admin OTRS	TicketCalendarEndTime	1 - Enabled
	TicketCalendarStartTime	1 - Enabled

Fig. 4: Change History Screen

39.3 External Interface

This package has no external interface.

System Monitoring

This module implements a basic interface to system monitoring suites.

Note: A network monitoring system, such as Nagios, Icinga2, HP OpenView or similar, capable of sending out events via email is required to use this package.

For Nagios, it works by receiving email messages sent by a network monitoring suite. New tickets are created in case of component failures. Once a ticket has been opened messages regarding the effected component are attached to this ticket. When the component recovers, the ticket state can be changed or the ticket can be closed. If an open ticket for a given host and service combination exists, all mails concerning this particular combination will be attached to the ticket until it' s closed.

For Icinga2, it works by specifying a host and a service in ticket dynamic fields. This combination of host and service is used after a ticket lock is set to generate a HTTP request which is send to the configured Icinga2 host (please see configuration chapter). In Icinga2, this request is used to create or to confirm new incidents.

Available in Service Package

- No service package needed

Package Name in OTRS Package Manager

- SystemMonitoring

40.1 Administrator Interface

This chapter describes the new features that are available in the administrator interface after installation of the package.

40.1.1 Control Flow for Nagios

The following control flow illustrates how mails are handled by this module and in which cases they trigger which action. Pretty much all checks are configurable using the regular expressions given by the parameters listed below.

- Mail matches `FromAddress?`
 - NO → Continue with regular mail processing
 - YES → Does a ticket with matching host and service combination already exist in **OTRS**?
 - * NO → Does `State: match NewTicketRegExp?`
 - NO → Stop processing this mail (silent drop)
 - YES → Create new ticket, record host and service, attach mail
 - * YES → Attach mail to ticket. Does `State: match CloseTicketRegExp?`
 - NO → Continue with regular mail processing
 - YES → Change ticket type as configured in `CloseActionState`

Besides of a few additional sanity checks this is how the system monitoring module treats incoming mails. By changing the regular expressions it should be possible to adopt it to different monitoring systems.

40.1.2 Icinga2 Acknowledge

A new ticket is created with values in the specified dynamic fields which are needed as combination of host and service for the communication to the Icinga2 host. After this newly created ticket is locked to an agent, a HTTP request is send to the configured Icinga2 host. In the Icinga2 host a new acknowledge is created or an existing one is confirmed.

Processes & Automation

After installation of the package some new dynamic fields will be added to the system, which are required for working this package properly.

Dynamic Fields

This package adds three new dynamic fields to the system.

Warning: Do not delete these dynamic fields. They are required to work the system monitoring package properly.

New Dynamic Fields

SystemMonitoring HostName This dynamic field will store the host name comes from the system monitoring suit.

- Internal name: `TicketFreeText1`
- Type: Text

- Object: Ticket

SystemMonitoring ServiceName This dynamic field will store the service name comes from the system monitoring suit.

- Internal name: `TicketFreeText2`
- Type: Text
- Object: Ticket

SystemMonitoring StateName This dynamic field will store the state name comes from the system monitoring suit.

- Internal name: `ArticleFreeText1`
- Type: Text
- Object: Article

40.2 Agent Interface

This package has no agent interface.

40.3 External Interface

This package has no external interface.

For service organizations that use the **OTRS** helpdesk software in different departments or that have to respond to many different types of customer requests, this feature add-on is a definite must-have. With it you can display different ticket masks and forms in the agent and external interfaces, depending on which dynamic fields are relevant to the customer request. Moreover, the ticket creation process is faster as headlines and message body fields are pre-filled with the necessary information. You can define required fields for forms as well as display pre-filled fields, such as ticket types and services, in the external interface based on the customer group settings. You can also pre-fill ticket masks by adding attachments, using Rich Text format, or using the OTRS smart tags `<OTRS_CONFIG_*>` and `<OTRS_CURRENT_*>` in the template body.

With this feature add-on the following screens can be configured:

- Agent interface
 - New phone ticket
 - New email ticket
- External interface
 - New ticket

In addition, the included feature add-on *Hide/Show Dynamic Fields*, which is also available as a separate feature add-on, enables you to define which dynamic fields you want to display in your ticket forms and which ones you want to hide. This gives you more freedom to configure all ticket screens to your specific needs.

Another included feature add-on is *Restore Pending Information*, which makes it possible to restore the pending reminder information of the last reminder when you open a new reminder in the ticket pending pop-up screen.

The following information can be restored:

- Pending date
- State
- Title
- Text (if the text comes from a note article created by the *Set Pending Time* action)

This feature add-on works when the target time of the reminder has not expired. Otherwise the standard functionality sets the usual default values.

Benefits

- Flexible reaction to a variety of customer requests.
- Acceleration of processes by using forms.
- Increases the completeness of forms by defining mandatory fields.

Target Groups

- Customer service/support
- IT service management
- Sales
- Public authorities

Available in Service Package

- GOLD

Package Name in OTRS Package Manager

- OTRSTicketForms

Note: Not compatible with the following feature add-ons:

- *Hide/Show Dynamic Fields*
 - *Restore Pending Information*
-

41.1 Administrator Interface

This package provides new features to the ticket screens, using the bundled packages *Hide/Show Dynamic Fields*, *Dynamic Ticket Templates* and *Restore Pending Information*, as well as other functionalities provided by this package itself.

Hide/Show Dynamic Fields package allows the definition of ACLs based visibility of dynamic fields. For more information see the documentation of the package.

Dynamic Ticket Templates package uses the extension to the ACLs mechanism to create ticket templates based on the selected ticket type and service. These templates include a predefined ticket body, title, attachments and full dynamic fields visibility configuration.

Restore Pending Information package provides the functionality to pre-select the target time for pending reminder with the last target time (including time, status, title and text) set by the agent. For more information see the documentation of the package.

41.1.1 Ticket Settings

After installation of the package a new module will be available in the *Ticket Settings* group of the administrator interface.

Dynamic Ticket Templates

Use this screen to add dynamic ticket templates for use in communications. The dynamic ticket template management screen is available in the *Dynamic Ticket Templates* module of the *Ticket Settings* group.

The screenshot shows the 'Manage Dynamic Ticket Templates' screen. On the left, there is a sidebar with an 'Add template' button and a filter box. Below the filter is a hint section. The main area contains a table with the following data:

NAME	ATTACHMENTS	COMMENT	VALIDITY	FRONTEND	CHANGED	CREATED	DEI
PeugeotTemplate	0	Peugeot Template	valid	Agent and Customer Interface	08/06/2019 15:50 (Europe/Budapest)	08/06/2019 15:50 (Europe/Budapest)	
VWTemplate1	0	VW Template 1	valid	Agent and Customer Interface	08/06/2019 15:47 (Europe/Budapest)	08/06/2019 15:47 (Europe/Budapest)	
VWTemplate2	0	VW Template 2	valid	Agent and Customer Interface	08/06/2019 15:48 (Europe/Budapest)	08/06/2019 15:48 (Europe/Budapest)	

Hint:
A dynamic ticket template is a default text and dynamic fields which helps your agents to write faster tickets, with a appropriate fields in the form.
Attention: Dynamic Fields are not needed to be setup in the screen configuration if they are set to be displayed for a template.

Fig. 1: Dynamic Ticket Templates Management Screen

Manage Dynamic Ticket Templates

Note: To add attachments to a dynamic ticket template, it needs to create the attachment first in the *Attachment Management* screen.

To add a dynamic ticket 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 dynamic ticket template:

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

To delete a dynamic ticket template:

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

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

Add Template

General:

★ Name:

Comment:

★ Valid:

★ Frontend:

Restrictions:

★ Type:

★ Service:

Template:

Workflow:

Subject:

Body:

B I U S | | | | | | | | | | | |

Format | Font | Size | **A** | **A** | **I_x** | Source

Attachments:

Brand:

VW Model:

VW Production Facility:

Peugeot Model:

Peugeot Production Facility:

Fuel:

Accessories:

Remarks:

Registration Date:

Invoice Date:

or Cancel

Edit Template

General:

★ Name: VWTemplate1

Comment: VW Template 1

★ Valid: valid

★ Frontend: Agent and External Interface

Restrictions:

★ Type: Unclassified

★ Service: VW Service x

Template:

Workflow: VWWorkflow1 x

Subject: VW Service Request 1

Body:

B I U S | | | | | |

Format | Font | Size | **A** **A** *I*_x | Source

Oil Change
Oil Filter Change
Air Filter Change
Fluids Check

Attachments:

Brand: Hide

VW Model: Show as mandatory

VW Production Facility: Show

Peugeot Model: Hide

Peugeot Production Facility: Hide

Fuel: Hide

Accessories: Hide

Remarks: Hide

Registration Date: Hide

Invoice Date: Hide

Save or **Save and finish** or Cancel




List							
NAME	ATTACHMENTS	COMMENT	VALIDITY	FRONTEND	CHANGED	CREATED	DELETE
PeugeotTemplate	0	Peugeot Template	valid	Agent and Customer Interface	08/06/2019 15:50 (Europe/Budapest)	08/06/2019 15:50 (Europe/Budapest)	
VWTemplate1	0	VW Template 1	valid	Agent and Customer Interface	08/06/2019 15:47 (Europe/Budapest)	08/06/2019 15:47 (Europe/Budapest)	
VWTemplate2	0	VW Template 2	valid	Agent and Customer Interface	08/06/2019 15:48 (Europe/Budapest)	08/06/2019 15:48 (Europe/Budapest)	

Fig. 4: Delete Dynamic Ticket Template Screen

Dynamic Ticket Template 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.

Valid * 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.

Frontend * Defines, where can the dynamic ticket template be used. A dynamic ticket template can be used in the following interfaces:

- Agent Interface
- Agent and External Interface
- External Interface

Type * Select a ticket type, that has been added to the system.

Service * Select a service, that has been added to the system.

Subject The subject of the email sent to the users.

Body The body of the email sent to the users.

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

Dynamic Fields The list of dynamic fields are displayed in this section. Choose for each dynamic field to *Hide*, *Show* or *Show as mandatory* on ticket create screens.

Dynamic Ticket Template Variables

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

Reference

You can use the following tags:

<OTRS_CURRENT_*>

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

<OTRS_CONFIG_*>

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

Example template:

```
From: "<OTRS_CURRENT_UserFirstname>"
Server: "<OTRS_CONFIG_FQDN>"
```

Fig. 5: Dynamic Ticket Template Variables

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
```

41.1.2 Communication & Notifications

This package adds a new ticket notification to the system.

Ticket Notifications

After installation of the package a new ticket notification will be added to the system.

New Ticket Notification

Missing DTT assignment for CustomerGroup If no template is assigned to a group and the notification is enabled, such notification can be customized by adding different recipients or updating the notification body.

To configure the optional settings for the notification:

1. Go to *System Configuration* screen.
2. Select *OTRSTicketForms* in the *Navigation* widget.
3. Navigate to *Core* → *DynamicTicketTemplate* in the navigation tree.
4. Enable `Ticket::DynamicTicketTemplate::CustomerGroup::Notify` setting to send a notification if a customer does not use a template to create a ticket.
5. Add agent login names to `Ticket::DynamicTicketTemplate::CustomerGroup::NotifyAgents` setting. The agent list in this setting will receive the notification when a user creates a ticket without using a template.

41.1.3 Users, Groups & Roles

After installation of the package a new module will be available in the *Users, Groups & Roles* group of the administrator interface.

Dynamic Ticket Templates Groups

Use this screen to add one or more dynamic ticket templates to one or more groups. To use this function, at least one dynamic ticket template and one group need to have been added to the system. The management screen is available in the *Dynamic Ticket Templates Groups* module of the *Users, Groups & Roles* group.

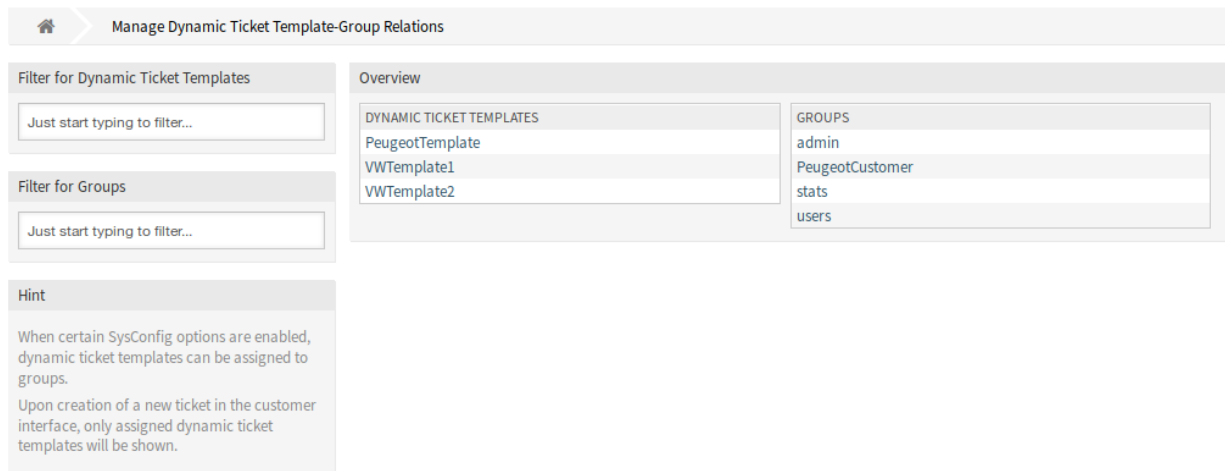


Fig. 6: Manage Dynamic Ticket Template-Group Relations

Manage Dynamic Ticket Templates Groups Relations

To assign some groups to a dynamic ticket template:

1. Click on a dynamic ticket template in the *Dynamic Ticket Templates* column.
2. Select the groups you would like to assign the dynamic ticket template to.
3. Click on the *Save* or *Save and finish* button.

To assign some dynamic ticket templates to a group:

1. Click on a group in the *Groups* column.
2. Select the dynamic ticket templates you would like to assign the group to.
3. Click on the *Save* or *Save and finish* button.

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

Multiple dynamic ticket templates or groups can be assigned in both screens at the same time. Additionally clicking on a dynamic ticket template or clicking on a group in the relations screen will open the *Edit Template* screen or the *Edit Group* screen accordingly.

Change Group Relations for Dynamic Ticket Template PeugeotTemplate

GROUPS	<input type="checkbox"/> ACTIVE
admin	<input type="checkbox"/>
PeugeotCustomer	<input checked="" type="checkbox"/>
stats	<input type="checkbox"/>
users	<input type="checkbox"/>

Save or **Save and finish** or Cancel

Fig. 7: Change Group Relations for Dynamic Ticket Template

Change Dynamic Ticket Template Relations for Group PeugeotCustomer

DYNAMIC TICKET TEMPLATES	<input type="checkbox"/> ACTIVE
PeugeotTemplate	<input checked="" type="checkbox"/>
VWTemplate1	<input type="checkbox"/>
VWTemplate2	<input type="checkbox"/>

Save or **Save and finish** or Cancel

Fig. 8: Change Dynamic Ticket Template Relations for Group

Warning: Accessing a dynamic ticket template or a group provides no back link to the relations screen.

41.1.4 Example Usage

For this example we will use the same fields as in the example of *Hide/Show Dynamic Fields*, but is necessary to comment or remove the ACLs.

Note: While the dynamic ticket templates can work in conjunction with other ACL rules, it is recommended to test the templates without other affecting ACLs, and after the templates work as needed, incorporate other ACLs one by one.

Preparation

To show the full functionalities of the dynamic ticket templates we can simply remove all dynamic fields from the *New Phone Ticket* screen by resetting the system configuration setting `Forms###AgentFrontend::TicketCreate::Phone::CreateProperties` to the default value, or disable the specific dynamic fields in the YAML configuration of this form:

```
- Label: Dynamic Fields
  Collapsible: 1
  Fields:
  - Name: DynamicField_Brand
    Inactive: 1
  - Name: DynamicField_VWModel
    Inactive: 1
  - Name: DynamicField_VWProductionFacility
    Inactive: 1
  - Name: DynamicField_PeugeotModel
    Inactive: 1
  - Name: DynamicField_PeugeotProductionFacility
    Inactive: 1
  - Name: DynamicField_Fuel
    Inactive: 1
  - Name: DynamicField_Accessories
    Inactive: 1
  - Name: DynamicField_Remarks
    Inactive: 1
  - Name: DynamicField_RegistrationDate
    Inactive: 1
  - Name: DynamicField_InvoiceDate
    Inactive: 1
```

To define a dynamic ticket template it is necessary to specify a ticket type and a service. If these features are not activated by default, please follow the next steps before start using this feature.

To activate ticket type and service features:

1. Go to *System Configuration* screen.
2. Select *OTRS* in the *Navigation* widget.
3. Navigate to *Core* → *Ticket* in the navigation tree.

4. Scroll down to the setting `Ticket::Service` and enable it.
5. Scroll down to the setting `Ticket::Type` and enable it.

Note: If you have `OTRSServiceManagement` package installed, these options are already active.

To activate ticket type and service features for the external interface:

1. Go to *System Configuration* screen.
2. Select *All settings* in the *Navigation* widget.
3. Navigate to *Frontend* → *External* → *View* → *TicketCreate* in the navigation tree.
4. Scroll down to the setting `ExternalFrontend::TicketCreate###Service` and enable it.
5. Scroll down to the setting `ExternalFrontend::TicketCreate###TicketType` and enable it.

Create the following services:

- *Peugeot Service*
- *VW Service*

All services should be available for the customer user who will use this feature, or marked as default services.

Goals of This Example

The goals of this example are the follows:

- If the selected service is *VW Service* and the selected dynamic ticket template is *VW Template 1*, the ticket form should look like:

Subject	VW Service Request 1
Body	<ul style="list-style-type: none"> – Oil Change – Oil Filter Change – Air Filter Change – Fluids Check
Dynamic Fields	<ul style="list-style-type: none"> – <code>VWModel</code>: visible and required – <code>VWProductionFacility</code>: visible

- If the selected service is *VW Service* and the selected dynamic ticket template is *VW Template 2*, the ticket form should look like:

Subject	VW Service Request 2
Body	The selected accessory reports the following issues:
Dynamic Fields	<ul style="list-style-type: none"> - VWModel: visible - Accessories: visible and required - Remarks: visible - RegistrationDate: visible - InvoiceDate: visible

- If the selected service is *Peugeot Service*, the ticket form should look like:

Subject	Peugeot Service Request
Body	The car computer reports the following issues:
Dynamic Fields	<ul style="list-style-type: none"> - PeugeotModel: visible - PeugeotProductionFacility: visible - Fuel: visible

Create Dynamic Ticket Templates

To create a dynamic ticket template:

1. Open the *Dynamic Ticket Templates* module of the *Ticket Settings* group in the administrator interface.
2. Click on the *Add template* button in the left sidebar.
3. Fill in the required fields.
4. Click on the *Save* button.

If you have added other dynamic fields by your own, the template form will show more fields than the ones explained in this examples. There is nothing to worry about, just leave all other dynamic fields as *Hide* while you create the new templates.

There are special dynamic fields from other packages that are not displayed in the template form. These special fields have a determined behavior and they should be always hidden, always shown or their visibility depends on other configurations, so they can not be part of the template definition. Also any dynamic field marked as *Internal* will not be displayed in the template form.

See also:

A quick way to know if a dynamic field is *Internal* is to look at dynamic fields in the overview table of the *Dynamic Fields Management* screen. *Internal* dynamic fields can not be deleted, and they does not have a icon in the *Delete* column.

Create the *VW Template 1* dynamic ticket template with the following data:

Field name	Value
Name	VWTemplate1
Comment	VW Template 1
Valid	valid
Frontend	Agent and External Interface
Type	Unclassified
Service	VW Service
Subject	VW Service Request 1
Body	<ul style="list-style-type: none"> • Oil Change • Oil Filter Change • Air Filter Change • Fluids Check
Attachments	
Brand	Hide
VWModel	Show as mandatory
VWProductionFacility	Show
PeugeotModel	Hide
PeugeotProductionFacility	Hide
Fuel	Hide
Accessories	Hide
Remarks	Hide
RegistrationDate	Hide
InvoiceDate	Hide

Create the *VW Template 2* dynamic ticket template with the following data:

Field name	Value
Name	VWTemplate2
Comment	VW Template 2
Valid	valid
Frontend	Agent and External Interface
Type	Unclassified
Service	VW Service
Subject	VW Service Request 2
Body	The selected accessory reports the following issues:
Attachments	
Brand	Hide
VWModel	Show
VWProductionFacility	Hide
PeugeotModel	Hide
PeugeotProductionFacility	Hide
Fuel	Hide
Accessories	Show as mandatory
Remarks	Show
RegistrationDate	Show
InvoiceDate	Show

Create the *Peugeot Template* dynamic ticket template with the following data:

Field name	Value
Name	PeugeotTemplate
Comment	Peugeot Template
Valid	valid
Frontend	Agent and External Interface
Type	Unclassified
Service	Peugeot Service
Subject	Peugeot Service Request
Body	The car computer reports the following issues:
Attachments	
Brand	Hide
VWModel	Hide
VWProductionFacility	Hide
PeugeotModel	Show
PeugeotProductionFacility	Show
Fuel	Show
Accessories	Hide
Remarks	Show
RegistrationDate	Show
InvoiceDate	Show

Activate Dynamic Ticket Templates

Once there are one or more templates activated in the system, you can use them.

Note: ACL restrictions will be ignored for the superuser account (UserID 1).

To activate a dynamic ticket template:

1. Click on the *Take Phone Call* menu item of the *Action* menu in the header bar.

There is a new field called *Dynamic Ticket Template*. By default it is empty and it will be filled automatically when you select the appropriate combination of ticket type and service.

2. Fill the form with the following data:

- Type: Unclassified
- Customer user: Customer 1
- Queue: Misc
- Service: VW Service

3. The *Dynamic Ticket Template* field will be automatically populated with *VW Template 1* and *VW Template 2*.
4. Select each template and match the resulting *New Phone Ticket* form with the expected results.
5. Now change the selected service to *Peugeot Service*. As there is only one template defined for this combination of ticket type and service, the template will be automatically selected for you.
6. Compare the resulting *New Phone Ticket* form with the expected results.

This complete example can be tested also *New Email Ticket* or external *New Ticket* screen without changing anything.

The template definition is independent from the screen and once a template is defined, it can be used on any ticket creation screen, but please review the dynamic field configuration on each screen for the default fields configuration when a template is not selected.

Dynamic Ticket Templates Customer Groups

For this example we will use the *Peugeot Template* defined in the example above.

The goal to have a pre-filled form to create a new ticket in external interface.

To configure a customer group:

1. Go to *System Configuration* screen.
2. Select *OTRS* in the *Navigation* widget.
3. Navigate to *Core* → *Customer* in the navigation tree.
4. Scroll down to the setting `CustomerGroupSupport` and enable it.
5. Select *OTRSTicketForms* in the *Navigation* widget.
6. Navigate to *Core* → *DynamicTicketTemplate* in the navigation tree.
7. Scroll down to the setting `Ticket::DynamicTicketTemplate::CustomerGroup` and enable it.
8. Go to the *Group Management* screen of the administrator interface.
9. Create the group *Peugeot Customer*.
10. Go to the *Customers Groups* screen of the administrator interface.
11. Assign *Customer 1* customer to the *Peugeot Customer* group.
12. Go to *Dynamic Ticket Templates Groups* screen of the administrator interface.
13. Assign the *Peugeot Customer* group to the *Peugeot Template* template.
14. Login in the external interface as customer user of *Customer 1* and create a new ticket.
15. Compare the pre-filled new ticket form with the expected results. Also notice that the ticket type is pre-selected as default and service is also pre-selected as *Peugeot Service*.

You can also set the optional settings, but for this example we will not use these optional settings, so you can leave it as default.

See also:

If no template is assigned to a group and the notification is enabled, such notification can be customized by adding different recipients or updating the notification body. To do this, edit the *Missing DTT assignment for CustomerGroup* notification in the *Ticket Notifications* screen of the administrator interface.

41.2 Agent Interface

This package has no agent interface.

See also:

There is a detailed use-case example in *Administrator Interface* chapter.

41.3 External Interface

This package has no external interface.

See also:

There is a detailed use-case example in *Administrator Interface* chapter.

This package adds the functionality of two new invokers `TicketCreate` and `TicketUpdate` for web services.

Available in Service Package

- GOLD

Package Name in OTRS Package Manager

- `OTRSTicketInvoker`

42.1 Administrator Interface

This package adds a new settings screen to the *Web Service Management* screen to configure the invokers.

42.1.1 Processes & Automation

This package adds two new invokers to *Invokers* section of the *Web Service Management* screen.

Web Services

After installation of the package, two new invokers will be available in the *Invokers* section. Selecting an invoker from the drop-down list will open a new settings window.

Request Example

`TicketCreate` and `TicketUpdate` invoker returns the complete ticket and article data based on ticket ID and article ID of the triggered event.

Prepare the invocation of the configured remote web service. Event data:

```

my $Result = $InvokerObject->PrepareRequest (
    Data => {                                     # data payload
        TicketID => 123,
        ArticleID => 123,                         # optional
    },
);

```

Invoker result:

```

{
  Data => {
    Ticket => {
      Title           => 'some ticket title',
      Queue           => 'some queue name',
      Lock             => 'some lock name',         # optional
      Type             => 'some type name',         # optional
      Service          => 'some service name',     # optional
      SLA              => 'some SLA name',         # optional
      State            => 'some state name',
      Priority         => 'some priority name',
      Owner            => 'some user login',       # optional
      Responsible      => 'some user login',       # optional
      CustomerUser     => 'some customer user login',
      PendingTime {   # optional
        Year   => 2011,
        Month  => 12,
        Day    => 03,
        Hour   => 23,
        Minute => 05,
      },
    },
    Article => {
      SenderType       => 'some sender type name', # optional
      AutoResponseType => 'some auto response type', # optional
      From              => 'some from string',     # optional
      Subject           => 'some subject',
      Body              => 'some body'
      ContentType       => 'some content type',     # ContentType or
↳MimeType and Charset is required
      MimeType          => 'some mime type',
      Charset           => 'some charset',
      TimeUnit          => 123,                     # optional
    },
    DynamicField => [                               # optional
      {
        Name   => 'some name',
        Value  => 'Value',                          # value type
↳depends on the dynamic field
      },
      # ...
    ],
    Attachment => [
      {

```

(continues on next page)

(continued from previous page)

```

        Content      => 'content'                # base64 encoded
        ContentType => 'some content type'
        Filename     => 'some fine name'
    },
    # ...
],
},
};

```

Note: The invoker will return the newest article of the ticket if no article ID is given.

Note: The invoker will not return dynamic fields with undefined values, because the same named operations `TicketCreate` and `TicketUpdate` does not handle dynamic fields with undefined values.

Advanced Filtering for Outgoing Data

For outgoing data, it is possible to define what kind of ticket fields, article fields or dynamic fields the request should contain. Furthermore it is possible to filter by article type and article sender type.

The different filter options can be selected within every single invoker configuration and are listed in section *Settings for outgoing request data*.

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

General Invoker Data

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 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.

Invoker backend This invoker back end module will be called to prepare the data to be sent to the remote system, and to process its response data. The field is read-only, it was selected in the previous screen.

Settings for Outgoing Request Data

Ticket fields A multi-select menu containing the available ticket attributes (fields), that can be submitted to a remote system. Just the fields that are selected will be included in outgoing requests.

Article fields A multi-select menu containing the available article attributes (fields), that can be submitted to a remote system. Just the fields that are selected will be included in outgoing requests.

Ticket dynamic fields A multi-select menu containing the available ticket dynamic fields, that can be submitted to a remote system. Just the dynamic fields that are selected will be included in outgoing requests.

Invoker Details

General invoker data★ Name:

The name is typically used to call up an operation of a remote web service.

Description: Invoker backend:

This OTRS invoker backend module will be called to prepare the data to be sent to the remote system, and to process its response data.

Settings for outgoing request dataTicket fields:

Only the selected ticket fields will be considered for the request data.

Article fields:

Only the selected article fields will be considered for the request data.

Ticket dynamic fields:

Only the selected ticket dynamic fields will be considered for the request data.

Article dynamic fields:

Only the selected article dynamic fields will be considered for the request data.

Number of articles:

The outgoing request data will only contain the configured number of articles. If left empty, only 1 article will be sent.

Communication channels:

The outgoing request data will only consider articles of the selected communication channels. If left empty, articles created by all communication channels will be used.

Customer visibility:

The outgoing request data will only consider articles created with the selected customer visibility.

Sender Types:

The outgoing request data will only consider articles created by the selected sender types. If left empty, articles created by all sender types will be used.

MappingMapping for outgoing request data:

The data from the invoker of OTRS will be processed by this mapping, to transform it to the kind of data the remote system expects.

Mapping for incoming response data:

The response data will be processed by this mapping, to transform it to the kind of data the invoker of OTRS expects.

Settings for incoming response dataRemote TicketID dynamic field:

Article dynamic fields A multi-select menu containing the available article dynamic fields, that can be submitted to a remote system. Just the dynamic fields that are selected will be included in outgoing requests.

Number of articles A text field containing the maximum number of articles, that will be transmitted during an outgoing request. The articles will be selected from newest (latest) to oldest. If no number is given, just the latest article will be transmitted.

Communication channels The outgoing request data will only consider articles of the selected communication channels. If left empty, articles created by all communication channels will be used.

Customer visibility The outgoing request data will only consider articles created with the selected customer visibility.

Sender Types The outgoing request data will only consider articles created by the selected sender types. If left empty, articles created by all sender types will be used.

Mapping

Mapping for outgoing request data The data from the invoker will be processed by this mapping, to transform it to the kind of data the remote system expects.

Mapping for incoming response data The response data will be processed by this mapping, to transform it to the kind of data the invoker expects.

Settings for Incoming Response Data

It is possible to automatically save certain data of incoming responses to local dynamic fields. The different filter options can be selected within every single invoker configuration.

Remote Ticket ID dynamic field A drop-down menu containing the available ticket dynamic fields in the system. If such a dynamic field is selected, the received ticket ID from the remote system will be used, to be saved within the selected dynamic field.

Ticket dynamic fields A multi-select menu containing the available ticket dynamic fields in the system. All selected dynamic fields, that also available in the response data and containing values, will be saved to the local dynamic fields.

The dynamic field values of the response data will be used from the following data structure:

```
<Ticket>
  <DynamicField>..</DynamicField>
</Ticket>
```

and/or

```
<Ticket>
  <Article>
    <DynamicField>..</DynamicField>
  </Article>
</Ticket>
```

The system configuration option `GenericInterface::Invoker::Settings::ResponseDynamicField` was added as a fallback for the dynamic fields that should contain the result ticket ID of the related response data. It shall be used, if the configuration was not added via the invoker GUI and both configurations should not be used at the same time!

Event Triggers

Event This invoker will be triggered by the configured events.

Add Event Trigger To add a new event select the event object and event name and click on the + button. Asynchronous event triggers are handled by the OTRS daemon in background (recommended). Synchronous event triggers would be processed directly during the web request.

Set Additional Outbound Headers

Note: This functionality is only available in combination with **STORM powered by OTRS™**.

This functionality allows to set arbitrary HTTP headers for outbound communication of `HTTP::REST` and `HTTP::SOAP` based web services (*Requester* requests and *Provider* responses), superseding the *additional response headers* functionality.

Within the *Transport* configuration it is now possible to define common headers (which are set for every outbound communication of the web service *Requester/Provider*) and *Invoker/Operation* specific headers (which are only set if the respective *Invoker/Operation* is used).

Headers are only allowed to be used once within a configuration type and those from specific configuration override common configuration headers.

Some headers are blocked for safety purposes. If needed, the list of blocked headers can be changed in system configuration using the settings `GenericInterface::Invoker::OutboundHeaderBlacklist` and `GenericInterface::Operation::OutboundHeaderBlacklist` respectively.

Note: Headers defined using the previous *additional response headers* functionality will continue to work and will be upgraded to the new functionality when the *Provider* transport configuration is changed the next time.

42.2 Agent Interface

This package has no agent interface.

42.3 External Interface

This package has no external interface.

Ticket Queue Selection

This feature add-on makes it possible to automatically move a ticket to preferred queues on the basis of ticket and customer data when it is created. In the standard version of **OTRS**, some similar selection can be done by using several email addresses or moving tickets manually.

After installing the feature add-on, you can define in the system configuration on which data the queue selection should be executed. For example, it would be possible to define certain customer names, which causes their tickets to move into a special queue where the appropriate customer consultant can work on it immediately.

It would also be possible to define keywords, like *problem* or *computer*, for the ticket title so that those tickets are automatically routed to the first-level-support queue.

Benefits

- Immediate handling of the ticket.
- Automatic assignment in the correct queue according to individually definable keywords.
- Saves time compared to a manual assignment.

Target Groups

- External IT service
- Call centers
- Sales
- Customer service
- And many more

Available in Service Package

- GOLD

Package Name in OTRS Package Manager

- OTRSTicketQueueSelection

43.1 Administrator Interface

This package has no administrator interface, but makes it possible to move tickets with specific ticket or customer user data into queues.

The package can be configured with the setting `OTRSTicketQueueSelection::Configuration`. `Key` is the queue name and value is a semicolon-separated list of ticket and/or customer user settings and their expected value as regular expression. The fields available for tickets and customer users have to match those available in **OTRS**.

43.1.1 Example Usage

This example shows how to configure a set of rules to move a ticket from any selected queue to a defined destination queue based on a ticket attribute, on ticket creation time.

To configure rules for newly created tickets with title `Test` to be moved to queue `Junk`:

1. Go to *System Configuration* screen.
2. Select `OTRSTicketQueueSelection` in the *Navigation* widget.
3. Navigate to *Core* → *Ticket* in the navigation tree.
4. Open `OTRSTicketQueueSelection::Configuration` setting for editing.
5. Enter `Junk` for key and `Ticket::Title='^Test$'` for content.
6. Create a ticket with title `Test` and check that the ticket is moved according to your configuration.

As seen in this example there is only one rule to that the ticket must match to be moved to the queue `Junk`. If more rules are needed, they can be added to the same content field with a `;` character as a separator, like:

```
Ticket::Title='^Test$';CustomerUser::UserFirstname='^John$';  
↪CustomerUser::UserLastname='^Doe$'
```

Each rule is taken by the system as an AND condition, so for these new rules the ticket will need to match the title condition **and** the customer user condition in order to be moved to the `Junk` queue.

The rules can be based on ticket attributes `Ticket::<Attribute>` or on customer user attributes `CustomerUser::<Attribute>`.

43.2 Agent Interface

This package has no agent interface.

43.3 External Interface

This package has no external interface.

Ticket Watchlist

This feature add-on allows you to manage several watchlists. A ticket agent defines watchlists and assigns tickets to a watchlist. Within the watchlist overview, the agent gets an overview of all of his watchlists and can administer them.

The feature add-on has the following features:

Define notification events for each watchlist The agent is able to administer the following events which trigger an email notification to himself:

- new article
- change of customer
- change of owner
- change of queue
- change of ticket status to a defined status

This email notification is independent of the notification preferences of the current owner of the ticket – unlike the standard subscribe/watchlist feature add-on of **OTRS**.

Assign a watchlist to another agent A watchlist can be handed over to another agent. The list will disappear in the overview of the former owner and appear in the overview of the new owner. This feature is typically used to hand the list over to another agent when the original agent goes on vacation.

Assign a deputy One or more deputy agents can be assigned to a watchlist. A deputy then sees the new watchlist in his overview, and he is allowed to add or remove tickets to or from this watchlist. This feature is typically used to share a watchlist with colleagues to work on a ticket in a team.

Export to CSV The list can be exported to a CSV file, e. g. to process it in Excel.

Benefits

- Get a quick and easy overview about all relevant tickets.
- Administration of several watchlists saves time.
- Export function enables the processing in Excel sheets.

Target Groups

- IT service management
- Service providers
- Call centers
- Service manager

Available in Service Package

- SILVER

Package Name in OTRS Package Manager

- OTRSTicketWatchlist

44.1 Administrator Interface

This package has no administrator interface.

The feature can be turned off with `Ticket::Watchlist` setting. In order to restrict this feature to specific groups, verify the `Ticket::WatchlistGroup` setting in the system configuration. By default, all users will have access to this feature.

It is possible to customize the watch list notifications in the *Ticket Notifications* screen.

44.1.1 Communication & Notifications

The watch list notifications are added to the system as regular ticket notifications.

Ticket Notifications

This package adds the following notifications to the system:

- Watchlist agent article notification
- Watchlist customer article notification
- Watchlist move notification
- Watchlist owner change notification
- Watchlist queue change notification
- Watchlist state change notification
- Ticket watcher reminder notification

To view the notifications:

1. Go to the *Ticket Notifications* screen of the administrator interface.
2. Find the notifications start with `Watchlist`.

44.2 Agent Interface

With this feature add-on the agents can create watchlists and they can be notified on various events.

A watchlist can contain unlimited number of tickets and a ticket can be assigned to unlimited number of watchlists.

44.2.1 Tickets

This package provides ticket watchlists and reminders for them in the ticket detail view, as well as adds a watchlist overview.

44.2.2 Ticket Detail View

Use this screen to add tickets to a watchlist or to remove tickets from a watchlist. A new menu item *Watchlists* is available in the breadcrumb bar.

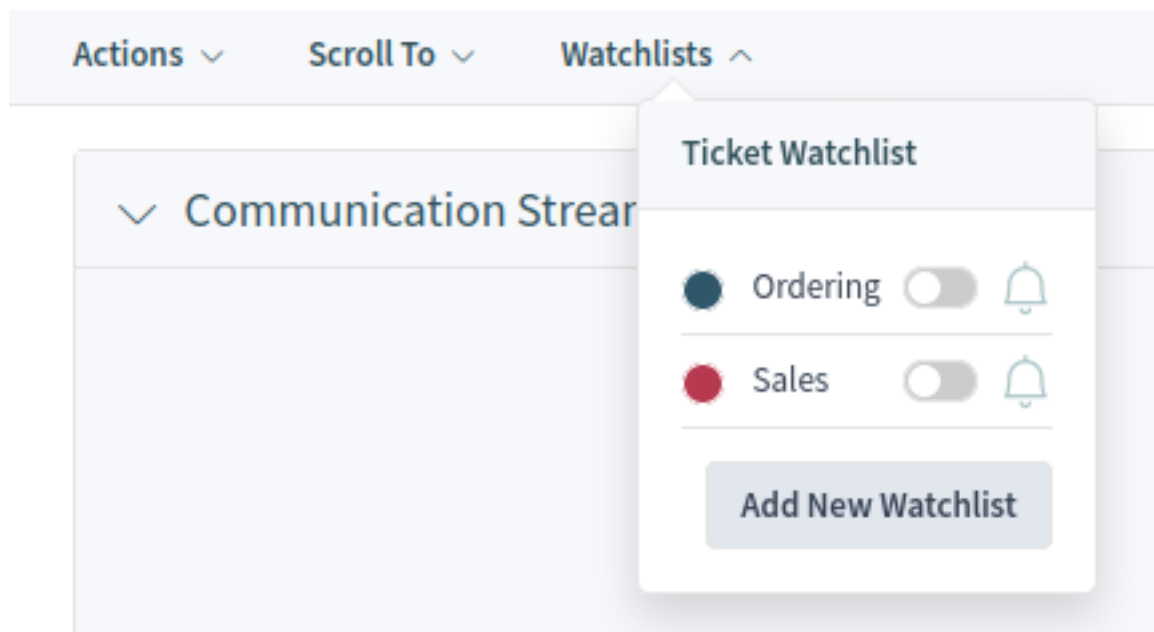


Fig. 1: Watchlists Menu

The ticket can be assigned to a watchlist via this menu item by using the toggle button. Additionally, the ticket can be removed from a watchlist. The *Add New Watchlist* button opens the *Ticket Watchlist Overview* screen and the *Create Ticket Watchlist* action.

If the ticket is added to the watchlist, the menu item will be changed to the color defined for the watchlist. If the ticket is added to multiple watchlists, all colors will be displayed.

Use the bell icon to set a reminder for the ticket in the watchlist. For this, the agent has to be subscribed to the ticket or it has to be assigned to one of his watchlists.

Note: This option is visible only, if the ticket is already added to a watchlist.

The *Watchlist Reminder* action has the following fields by default:

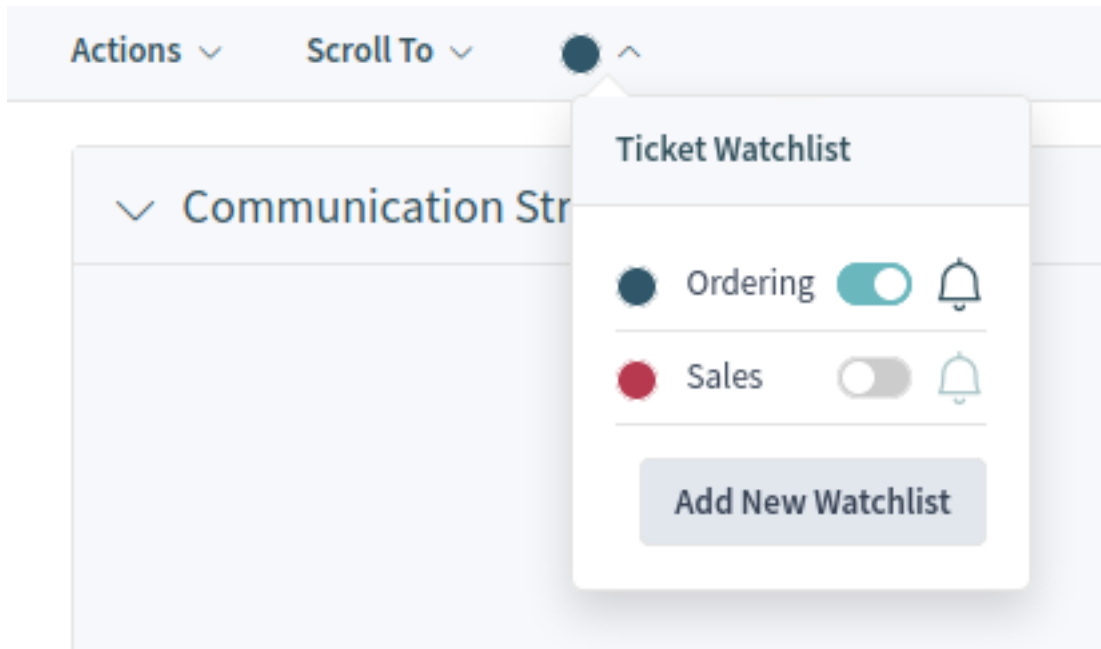


Fig. 2: Added Ticket to Watchlist

Reminder active Controls, whether the reminder is active or not. New reminders will be added only if this checkbox is checked. For existing reminders, unchecking the checkbox will remove the reminder.

Reminder Time * Set the date and time when the reminder should be displayed.

Reminder for * Select a watchlist from the drop-down list.

Note Add additional information to the reminder text.

Note: There can only be one active reminder per agent and ticket. Notifications are only being sent every ten minutes.

If a reminder is defined for the ticket in a watchlist, the bell icon will be filled.

A watchlist belongs to the agent who created it. The list can be moved to another agent in the [Ticket Watchlist Overview](#) screen. When moving a watchlist to another agent, reminders will also be moved.

In this case, where the owner of the watchlist is changed, the notification that is sent is different that the others. It can only be sent by the email notification method and it should not contain any ticket related OTRS smart tag (if so they will be removed before sending) as there is no ticket involved in the process that triggers the event.

The default *Watchlist move notification* is attached to the event `WatchlistMoveNotification`. This and any other ticket notification attached to the same event should prevent the use of these ticket related OTRS smart tags.

Reminders will only be sent if the assigned agent at this time still has access to the watchlist containing the ticket and/or still is subscribed to the ticket.

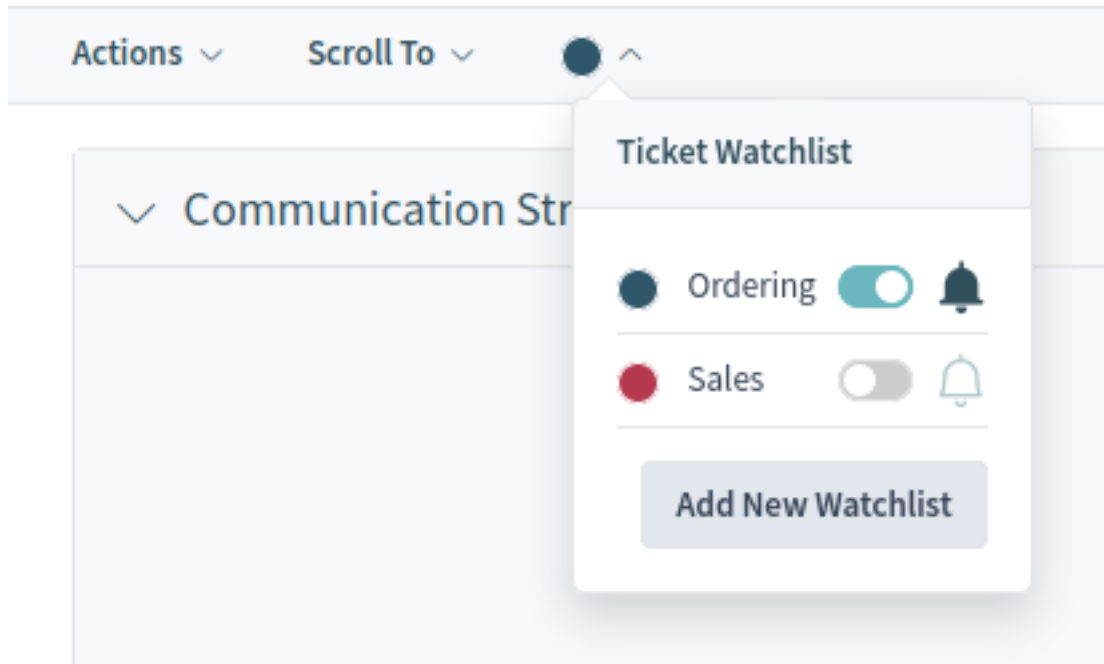


Fig. 3: Active Watchlist Reminder

44.2.3 Ticket Watchlist Overview

Use this screen to manage watchlists. The ticket watchlists overview screen is available from the organizer via the glasses icon.

Manage Watchlists

To add a watchlist:

1. Click on the plus icon in the breadcrumb bar.
2. Fill in the required field.
3. Click on the *Save* button.

To export the ticket list from a watchlist:

1. Click on the export button in the *Actions* column of a watchlist.
2. Select the export format.
3. Click on the *Download File* button.
4. Save the file somewhere in the file system.

To edit a watchlist:

1. Click on the edit button in the *Actions* column of a watchlist.
2. Modify the field.
3. Click on the *Save* button.

To change the watchlist owner:

1. Click on the person icon in the *Actions* column of a watchlist.

2. Select an agent from the list.
3. Click on the *Save* button.

Warning: The original owner agent will not have access to the watchlist anymore.

To assign deputies to a watchlist:

1. Click on the police badge icon in the *Actions* column of a watchlist.
2. Select deputy agents from the list.
3. Click on the *Save* button.

To delete a watchlist:

1. Click on the trash icon in the *Actions* column of a watchlist.
2. Click on the *Yes* button in the confirmation dialog.

Tickets can be added to the watchlist or removed from the watchlist in the *Tickets* screen.

Watchlist 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.

Color * The color associated to the watchlist. This color will be displayed in the ticket detail view when a watchlist is selected for the ticket.

Notify On Select the events that will trigger a notification.

New Agent Article Select, if the watchlist owner should be notified, when an agent adds a note to a ticket added to the watchlist.

New Customer Article Select, if the watchlist owner should be notified, when a customer user creates an article in the external interface.

New Owner Select, if the watchlist owner should be notified, when a ticket added to the watchlist has a new owner.

New Queue Select, if the watchlist owner should be notified, when a ticket added to the watchlist has been moved to an other queue.

New State Select, if the watchlist owner should be notified, when a state is changed for a ticket added to the watchlist.

Target States Notifications for status changes are only being sent if the new status matches the configured destination status.

If the *New State* notification above is selected, this field becomes mandatory.

Watchlist owners and their deputies can configure the notifications separately. The configuration of the destination status is only allowed for the watchlist owner and therefore also affects the deputies.

Notifications are not being sent to the agent who triggered the action.

See also:

The content of the notifications can be edited in the ticket detail view when the ticket is assigned to a watchlist. See the [Tickets](#) chapter for more details.

44.3 External Interface

This package has no external interface.