



OTRS Feature Add-ons Manual

Release 7.0

OTRS AG

Jan 20, 2021

Contents

1	Introduction	3
1.1	Categories	3
1.2	Service Levels	5
2	Adjust Sort Ticket Overview	9
2.1	Administrator Interface	9
2.2	Agent Interface	11
2.3	External Interface	11
3	Advanced Editor	13
3.1	Administrator Interface	14
3.2	Agent Interface	16
3.3	External Interface	16
4	Advanced Escalations	17
4.1	Administrator Interface	18
4.2	Agent Interface	34
4.3	External Interface	38
5	Advanced Generic Agent	39
5.1	Administrator Interface	40
5.2	Agent Interface	40
5.3	External Interface	40
6	Agent Email Interface	41
6.1	Administrator Interface	42
6.2	Agent Interface	42
6.3	External Interface	43
7	Automated FAQ Ticket Creator	45
7.1	Administrator Interface	46
7.2	Agent Interface	51
7.3	External Interface	53
8	Automatic Start of Processes	55
8.1	Administrator Interface	56
8.2	Agent Interface	57

8.3	External Interface	58
9	Brute Force Attack Protection	59
9.1	Administrator Interface	59
9.2	Agent Interface	62
9.3	External Interface	62
10	Calendar Resource Planning	63
10.1	Administrator Interface	64
10.2	Agent Interface	68
10.3	External Interface	78
11	Categories for Text Modules	79
11.1	Administrator Interface	80
11.2	Agent Interface	84
11.3	External Interface	86
12	Change Write Protection	87
12.1	Administrator Interface	88
12.2	Agent Interface	88
12.3	External Interface	89
13	CI Assignment Attribute Dynamic Field Map	91
13.1	Administrator Interface	92
13.2	Agent Interface	94
13.3	External Interface	94
14	CI Custom Search	95
14.1	Administrator Interface	96
14.2	Agent Interface	96
14.3	External Interface	97
15	CI References	99
15.1	Administrator Interface	100
15.2	Agent Interface	103
15.3	External Interface	104
16	CIs in Customer Frontend	105
16.1	Administrator Interface	106
16.2	Agent Interface	108
16.3	External Interface	109
17	Credit Card Filter	113
17.1	Administrator Interface	114
17.2	Agent Interface	115
17.3	External Interface	117
18	Custom Contact Fields	119
18.1	Administrator Interface	120
18.2	Agent Interface	123
18.3	External Interface	124
19	Customer Activity Widget	125
19.1	Administrator Interface	126
19.2	Agent Interface	126
19.3	External Interface	127

20 Customer Event Ticket Calendar	129
20.1 Administrator Interface	129
20.2 Agent Interface	130
20.3 External Interface	130
21 Customer Frontend Link Object	133
21.1 Administrator Interface	133
21.2 Agent Interface	134
21.3 External Interface	134
22 Customer-specific Services	135
22.1 Administrator Interface	136
22.2 Agent Interface	139
22.3 External Interface	139
23 Dashboard News Widget	141
23.1 Administrator Interface	141
23.2 Agent Interface	144
23.3 External Interface	146
24 Data Privacy Protection	147
24.1 Administrator Interface	148
24.2 Agent Interface	162
24.3 External Interface	162
25 Delete Attachments	163
25.1 Administrator Interface	163
25.2 Agent Interface	165
25.3 External Interface	165
26 Dynamic Field Attachment	167
26.1 Administrator Interface	167
26.2 Agent Interface	169
26.3 External Interface	170
27 Dynamic Field Calculation	171
27.1 Administrator Interface	171
27.2 Agent Interface	172
27.3 External Interface	173
28 Dynamic Field CI	175
28.1 Administrator Interface	176
28.2 Agent Interface	179
28.3 External Interface	179
29 Dynamic Field Database	181
29.1 Administrator Interface	182
29.2 Agent Interface	185
29.3 External Interface	186
30 Dynamic Field Search Booster	187
30.1 Administrator Interface	188
30.2 Agent Interface	188
30.3 External Interface	188

31 Dynamic Field Value Import	189
31.1 Administrator Interface	190
31.2 Agent Interface	194
31.3 External Interface	194
32 Dynamic Field Web Service	195
32.1 Administrator Interface	196
32.2 Agent Interface	204
32.3 External Interface	204
33 Dynamic Sender Addresses	205
33.1 Administrator Interface	206
33.2 Agent Interface	209
33.3 External Interface	211
34 Escalation Suspend	213
34.1 Administrator Interface	214
34.2 Agent Interface	214
34.3 External Interface	215
35 Extended Ticket Stats	217
35.1 Administrator Interface	218
35.2 Agent Interface	218
35.3 External Interface	221
36 Hide/Show Dynamic Fields	223
36.1 Administrator Interface	224
36.2 Agent Interface	230
36.3 External Interface	230
37 Link Object Connector	231
37.1 Administrator Interface	231
37.2 Agent Interface	240
37.3 External Interface	240
38 Out of Office	241
38.1 Administrator Interface	242
38.2 Agent Interface	242
38.3 External Interface	242
39 Process Management Article Email	243
39.1 Administrator Interface	244
39.2 Agent Interface	245
39.3 External Interface	245
40 Queue Responsible	247
40.1 Administrator Interface	247
40.2 Agent Interface	248
40.3 External Interface	248
41 Ready2Adopt ITSM Processes	249
41.1 Administrator Interface	250
41.2 Agent Interface	251
41.3 External Interface	251

42 Ready2Adopt Processes	253
42.1 Administrator Interface	254
42.2 Agent Interface	254
42.3 External Interface	254
43 Ready2Adopt Web Services	255
43.1 Administrator Interface	256
43.2 Agent Interface	256
43.3 External Interface	256
44 Restore Pending Information	257
44.1 Administrator Interface	258
44.2 Agent Interface	258
44.3 External Interface	259
45 Restrict Customer Data View	261
45.1 Administrator Interface	262
45.2 Agent Interface	264
45.3 External Interface	264
46 Service-based Queue Routing	265
46.1 Administrator Interface	266
46.2 Agent Interface	268
46.3 External Interface	268
47 Service Categories	269
47.1 Administrator Interface	270
47.2 Agent Interface	272
47.3 External Interface	272
48 Service Responsible	273
48.1 Administrator Interface	274
48.2 Agent Interface	275
48.3 External Interface	276
49 Specific Ticket Notifications	277
49.1 Administrator Interface	278
49.2 Agent Interface	280
49.3 External Interface	282
50 State Pre-selection Response Templates	283
50.1 Administrator Interface	284
50.2 Agent Interface	285
50.3 External Interface	285
51 System Configuration History	287
51.1 Administrator Interface	287
51.2 Agent Interface	289
51.3 External Interface	291
52 Tag Cloud	293
52.1 Administrator Interface	294
52.2 Agent Interface	294
52.3 External Interface	297

53 Ticket Allocation	299
53.1 Administrator Interface	300
53.2 Agent Interface	304
53.3 External Interface	304
54 Ticket Forms	305
54.1 Administrator Interface	306
54.2 Agent Interface	321
54.3 External Interface	321
55 Ticket Invoker	323
55.1 Administrator Interface	323
55.2 Agent Interface	328
55.3 External Interface	328
56 Ticket Queue Selection	329
56.1 Administrator Interface	330
56.2 Agent Interface	330
56.3 External Interface	330
57 Ticket Time Unit Dropdown	331
57.1 Administrator Interface	332
57.2 Agent Interface	332
57.3 External Interface	332
58 Ticket Watchlist	333
58.1 Administrator Interface	334
58.2 Agent Interface	335
58.3 External Interface	340
59 Ticket Workflow	341
59.1 Administrator Interface	342
59.2 Agent Interface	349
59.3 External Interface	350
60 Time Accounting Quota	351
60.1 Administrator Interface	352
60.2 Agent Interface	352
60.3 External Interface	354
61 VIP Customer	355
61.1 Administrator Interface	356
61.2 Agent Interface	356
61.3 External Interface	356

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

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](#).

1.1 Categories

Each feature add-on can be categorized based on its purpose.

Administration

- *Dynamic Field Value Import*
- *System Configuration History*

Automation & Processes

- *Advanced Editor*
- *Advanced Generic Agent*
- *Agent Email Interface*
- *Automated FAQ Ticket Creator*
- *CI Assignment Attribute Dynamic Field Map*
- *Dynamic Sender Addresses*
- *Process Management Article Email*
- *Queue Responsible*

- *Ready2Adopt ITSM Processes*
- *Ready2Adopt Processes*
- *Service-based Queue Routing*
- *Service Responsible*
- *Ticket Allocation*
- *Ticket Queue Selection*
- *Ticket Workflow*

Customer Management

- *Custom Contact Fields*
- *Customer Activity Widget*
- *Customer-specific Services*
- *Time Accounting Quota*
- *VIP Customer*

Individualization

- *Adjust Sort Ticket Overview*
- *CI Custom Search*
- *Hide/Show Dynamic Fields*
- *Specific Ticket Notifications*
- *Ticket Forms*

Integration

- *Dynamic Field Database*
- *Dynamic Field Web Service*
- *Link Object Connector*
- *Ready2Adopt Web Services*
- *Ticket Invoker*

Knowledge Management & Self Service

- *CIs in Customer Frontend*
- *Customer Event Ticket Calendar*
- *Customer Frontend Link Object*
- *Dashboard News Widget*
- *CI References*
- *Tag Cloud*

Performance

- *Dynamic Field Search Booster*

Reporting

- *Extended Ticket Stats*

Security & Permissions

- *Change Write Protection*
- *Credit Card Filter*
- *Data Privacy Protection*
- *Delete Attachments*
- *Restrict Customer Data View*

Ticket Management

- *Categories for Text Modules*
- *Dynamic Field Attachment*
- *Dynamic Field Calculation*
- *Dynamic Field CI*
- *Out of Office*
- *Restore Pending Information*
- *Service Categories*
- *State Pre-selection Response Templates*
- *Ticket Watchlist*

Time Management

- *Advanced Escalations*
- *Escalation Suspend*
- *Ticket Time Unit Dropdown*

1.2 Service Levels

Not all feature add-ons are available on any service levels. The following categorization is based on service levels.

SILVER

- *Credit Card Filter*
- *Calendar Resource Planning*
- *Custom Contact Fields*
- *Customer Activity Widget*
- *Customer Frontend Link Object*
- *Customer-specific Services*
- *Dashboard News Widget*
- *Delete Attachments*
- *Dynamic Field Database*
- *Ready2Adopt Processes*
- *Restore Pending Information*

- *Service-based Queue Routing*
- *Service Categories*
- *State Pre-selection Response Templates*
- *System Configuration History*
- *Tag Cloud*
- *Ticket Time Unit Dropdown*
- *Ticket Watchlist*
- *Ticket Workflow*
- *Time Accounting Quota*
- *VIP Customer*

GOLD

- *Adjust Sort Ticket Overview*
- *Advanced Editor*
- *Advanced Generic Agent*
- *Agent Email Interface*
- *Categories for Text Modules*
- *Change Write Protection*
- *CI Assignment Attribute Dynamic Field Map*
- *CI Custom Search*
- *CI References*
- *CIs in Customer Frontend*
- *Dynamic Field Attachment*
- *Dynamic Field Calculation*
- *Dynamic Field CI*
- *Dynamic Field Value Import*
- *Dynamic Field Web Service*
- *Dynamic Sender Addresses*
- *Escalation Suspend*
- *Hide/Show Dynamic Fields*
- *Link Object Connector*
- *Process Management Article Email*
- *Queue Responsible*
- *Ready2Adopt Web Services*
- *Restrict Customer Data View*
- *Specific Ticket Notifications*
- *Ticket Allocation*

- *Ticket Forms*
- *Ticket Invoker*
- *Ticket Queue Selection*

TITANIUM

- *Automated FAQ Ticket Creator*
- *Customer Event Ticket Calendar*
- *Data Privacy Protection*
- *Out of Office*
- *Ready2Adopt ITSM Processes*
- *Service Responsible*

PLATINUM

- *Advanced Escalations*
- *Dynamic Field Search Booster*
- *Extended Ticket Stats*

Adjust Sort Ticket Overview

This feature add-on makes it possible to modify the columns of the small ticket overview screens and the dashboard. It offers the following customization options:

- Definition of a default width of columns.
- Configurable word length of title column.
- Configurable width of columns via drag and drop on a per-user basis.
- Setting the queue and service display names.
- Make ticket number column always visible.

Benefits

- Facilitates work with the small agent ticket overview.
- Increases flexibility through optimized adaptability.

Target Groups

- IT service management
- Internal IT service
- Customer service/support

Available in Service Package

- GOLD

Package Name in OTRS Package Manager

- OTRSAdjustSortTicketOverview

2.1 Administrator Interface

This package has no administrator interface.

2.1.1 System Configuration

Width of columns The width of the columns in the small ticket overviews are configurable.

To configure the width of columns:

1. Go to *System Configuration* screen.
2. Select *OTRSAdjustSortTicketOverview* in the *Navigation* widget.
3. Navigate to *Frontend* → *Agent* → *TicketOverview* in the navigation tree.
4. Add new entries to setting `Frontend::Agent::TicketOverview::OTRSAdjustSortTicketOverviewFix`
Examples:

- Queue → *10%*
- State → *8%*
- Age → *40px*
- `DynamicField_DateofReceipt` → *3cm*

Configurable fields are: `TicketNumber`, `Age`, `EscalationTime`, `Title`, `State`, `Queue`, `Lock`, `Owner`, `CustomerID`, `DynamicField_*`.

This feature is activated by default after installation of the package.

Word length of title column The word length of the title column in the ticket overview widgets is configurable.

To configure the word length:

1. Go to *System Configuration* screen.
2. Select *OTRSAdjustSortTicketOverview* in the *Navigation* widget.
3. Navigate to *Frontend* → *Agent* → *TicketOverview* in the navigation tree.
4. Set the word length in setting `Frontend::Agent::TicketOverview::OTRSAdjustSortTicketOverviewW`

Drag and drop table column widths The width of the columns in the ticket overview widgets and the dashboards are configurable via drag and drop on a per-user-base.

To activate this feature:

1. Go to *System Configuration* screen.
2. Select *OTRSAdjustSortTicketOverview* in the *Navigation* widget.
3. Navigate to *Frontend* → *Agent* → *TicketOverview* in the navigation tree.
4. Enable setting `Frontend::Agent::TicketOverview::OTRSAdjustSortTicketOverviewUseDragAndD`

After activation it is possible to drag and drop the different table columns with the mouse to set the widths of every single column. Those settings are saved and restored for every single user in the user preferences.

Queue and service display names Depending on a system configuration setting, queue and service names can be displayed with complete paths, only with first letter of parents followed by `::` and their names, or just by their names in the ticket overview widgets.

To modify the display names:

1. Go to *System Configuration* screen.
2. Select *OTRSAdjustSortTicketOverview* in the *Navigation* widget.

3. Navigate to *Frontend* → *Agent* → *TicketOverview* in the navigation tree.
4. Set the display names in setting `Frontend::Agent::TicketOverview::ProcessQueueServiceFields`.

Warning: There is a potential risk that if ticket title (if double colons are contained) is also displayed in shortened manner because internally the double colons are used for creating the correct shortened path for queue and service name.

Ticket number always visible This setting prevents the users to hide the ticket number column by drag and drop.

To activate this feature:

1. Go to *System Configuration* screen.
2. Select *OTRSAdjustSortTicketOverview* in the *Navigation* widget.
3. Navigate to *Frontend* → *Agent* → *TicketOverview* in the navigation tree.
4. Enable setting `Frontend::Agent::TicketOverview::TicketNumberAlwaysVisible`.

Affected screens The features can be enabled or disabled in each ticket overview screens.

To modify the affected screens:

1. Go to *System Configuration* screen.
2. Select *OTRSAdjustSortTicketOverview* in the *Navigation* widget.
3. Navigate to *Frontend* → *Base* → *OutputFilter* in the navigation tree.
4. Add or remove screen names in the `Template` key of each setting.

2.2 Agent Interface

This package has no agent interface, only the existing ticket overview widgets get new features in the following screens:

- Agent Dashboard
- Small Queue View
- Small Status View
- Small Escalation View
- Customer Information Center
- Customer User Information Center

The new features are described in the [Administrator Interface](#) chapter.

2.3 External Interface

This package has no external interface.

Advanced Editor

This feature add-on allows OTRS administrators and OTRS developers with experience using Template Toolkit to write Template Toolkit code snippets in OTRS templates. This can facilitate, for example, the customization of response templates with multiple languages because you can customize the same template for many languages by using conditions without adding completely new response templates. This saves resources within **OTRS** and gives you a cleaner overview of your response templates.

Benefits

- Easier customization of response templates based on conditions.
- Resource-saving usage of response templates.
- Increases your **OTRS** performance.

Target Groups

- OTRS administrators with Template Toolkit experience
- Organizations that want to easily customize their OTRS templates, e.g. for many languages

Available in Service Package

- GOLD

Package Name in OTRS Package Manager

- OTRSAdvancedEditor

Note: Not compatible with the following feature add-ons:

- *Categories for Text Modules*
 - *Hide/Show Dynamic Fields*
 - *Ticket Forms*
-

3.1 Administrator Interface

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

3.1.1 Create New Templates

This feature enables logical if-else conditions as well as loops and much more.

To insert code snippets:

1. Open *Templates*, *Signatures* or *Salutations* modules in the administrator interface.
2. Add or modify a resource in these screens.
3. Find the new icon in the Rich Text editor toolbar (the last one).

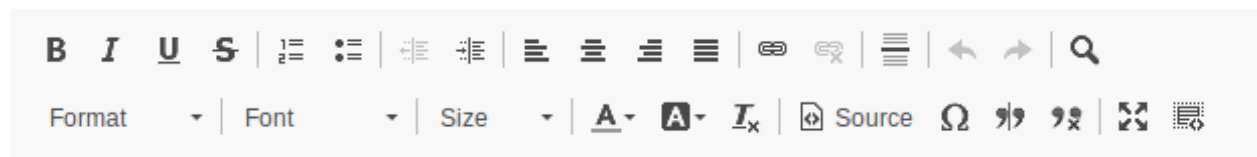


Fig. 1: Insert Code Snippet Icon

4. Click on the new icon to open a new window.
5. Select *TemplateToolkit* from the *Language* drop-down.
6. Write the following lines into the text box.

```
[% IF ( matches = Data.Language.match('^de') ) %]
Sehr geehrte(r) [% Data.CustomerUser.UserFirstname %] [% Data.
→CustomerUser.UserLastname %], <br />
[% ELIF ( matches = Data.Language.match('^hu') ) %]
Tisztelt [% Data.CustomerUser.UserLastname %] [% Data.CustomerUser.
→UserFirstname %]!<br />
[% ELSE %]
Dear [% Data.CustomerUser.UserFirstname %] [% Data.CustomerUser.
→UserLastname %], <br />
[% END %]
```

7. Click on the *OK* button to save the code snippet.
8. Click on the *Save* or *Save and finish* button.
9. Add the template, signature or salutation to the queue you want to use it.
10. Create a new ticket, choose a customer user (for this example we will call him John Doe), then choose the queue you have configured your template for.

If the customer user language is German, it should print *Sehr geehrte(r) John Doe,*. If it is set to Hungarian, you should see *Tisztelt Doe John!*. If your customer user has no language set or it is set to English, you will see *Dear John Doe,*.

As you can see, conditions in templates can be very useful, but be careful to write correct Template Toolkit syntax. If you make mistakes there, you will get an error during choosing a template.

See also:

For detailed instructions on how to write Template Toolkit code, refer to the [Template Toolkit User Manual](#).

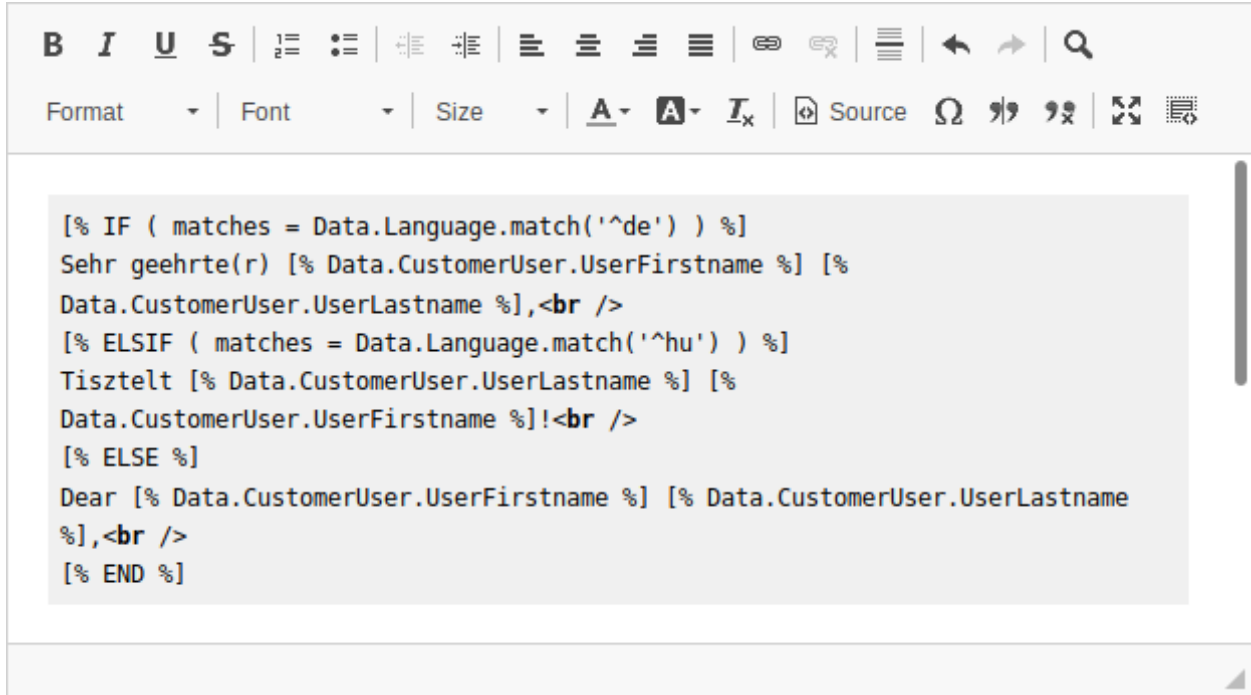
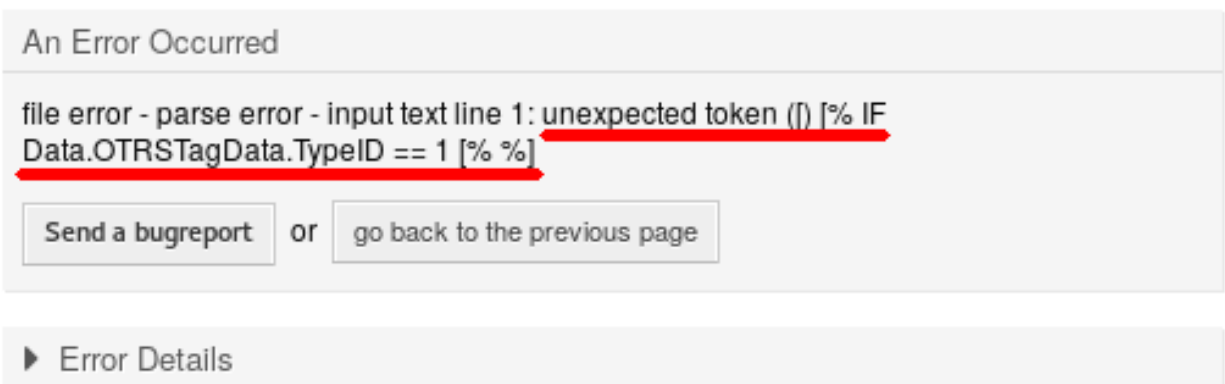


Fig. 2: Rich Text Editor Code Snippet

3.1.2 Debugging Templates

Writing Template Toolkit code is not too difficult compared to other languages. Nevertheless errors can occur as we are all just humans. To validate your template, open the ticket create or answer screen, where you have your templates configured to be used and choose a template.

If you have mistakes in your Template Toolkit syntax, choosing a template will bring up an error screen that tell you about what went wrong and where to find the error.



Error Reason in Error Screen

In this example the **OTRS** system informed us, that there was an unexpected token [in the line [% IF Data.OTRSTagData.TypeID == 1 [% %]. A second opening bracket was entered inside the same Template Toolkit tag, which for sure is forbidden.

To fix the error:

1. Go back to the template editing screen.
2. Edit the template and save it.
3. Start debugging the template again by creating or answering a ticket, till no more errors are found and your template comes up as desired.

3.2 Agent Interface

This package has no agent interface.

3.3 External Interface

This package has no external interface.

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 view, the new widget *Escalation Information* appears. This shows through the use of different colors and numerical values whether or not:

- the escalation time is still within the original timeframe (green),
- the escalation time will run out soon (yellow),
- the escalation is paused (brown),
- 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 window 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

Note: Not compatible with the following feature add-ons:

- *Escalation Suspend*
 - *Ticket Watchlist*
-

4.1 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.

4.1.1 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.

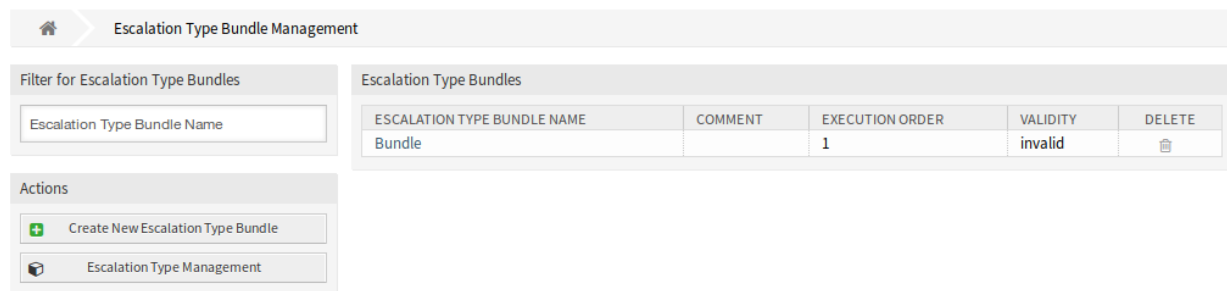


Fig. 1: Escalation Type Bundle Management Screen

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

One or more 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

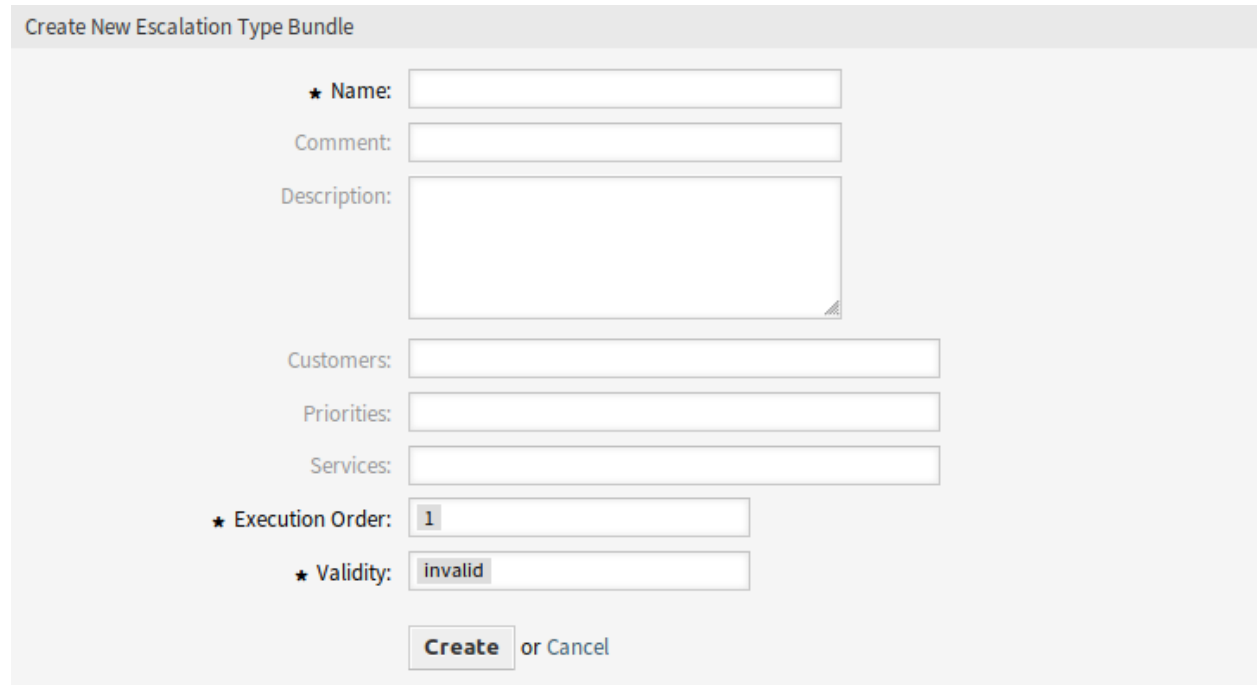
To create a new escalation type bundle:

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:



Create New Escalation Type Bundle

★ Name:

Comment:

Description:

Customers:

Priorities:

Services:

★ Execution Order:

★ Validity:

or

Fig. 2: Create New Escalation Type Bundle Screen

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

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.

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

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.

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.

The screenshot shows a web form titled "Create New Escalation Type". It contains the following elements:

- A text input field for "★ Name:".
- A text input field for "Comments:".
- A larger text area for "Description:".
- A checkbox labeled "Allow resume from stop:" which is currently unchecked.
- A dropdown menu for "★ Validity:" with "invalid" selected.
- Two buttons at the bottom: "Create" and "Cancel".

Fig. 6: Create New Escalation Type Screen

To edit an escalation type:

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.

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"/> <input type="text"/>
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"/> <input type="text"/>

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

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.

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 Cancel

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.

4.1.2 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###TriggerEscalation` 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.

4.1.3 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

Visualization of Escalation Times in Ticket Overviews

In the ticket overviews and also on ticket dashboard widgets, information regarding the escalation data is displayed in one column for each escalation type.

In addition to this, you have the possibility to specify which advanced escalations columns you want to use in ticket overviews and ticket dashboard widgets.

In the system configuration options the possible advanced escalations columns are predefined, but they are not active. If you want to use them, you have to change the value to 1 in setting `AgentDashboard###AdvancedEscalationsDefaultColumns`.

Due to a technical reason there is only one system configuration option for the ticket dashboard widgets. So you are not able to define the advanced escalations columns for each ticket dashboard widget.

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

For the user based settings the modal dialog for the configuration of views has been extended to offer a selection of available escalation types.

The traffic light colors for the escalation states:

- Green: No escalation times are reached or exceeds the limit.
- Yellow/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.
- Brown: Escalation time is currently paused.

Possible Advanced Escalations Columns

For each configured escalation type the following 22 advanced escalations columns are calculated and available. It is not necessary to activate the columns like you have to do e.g. for the agent dashboard. 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).

4.1.4 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.

4.2 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 overviews, ticket dashboard 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.

4.2.1 Tickets

After installation of the package a new *Escalation Information* widget is displayed in *Ticket Zoom* screen.

Ticket Zoom

Display Escalations in Overviews

All advanced escalations that are associated to a ticket will be shown in *Escalation Information* widget in *Ticket Zoom* screen.

The escalations are displayed with a colored background indicating the current escalation state for better visibility.

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

Debug escalation

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.

History of Ticket#: 201502203549000024 - test
Close

OTRSAdvancedEscalations - EscalationType Debug Information

First Response
Update Time
Solution Time

ACTION	COMMENT	ZOOM	USER	CREATETIME
NewTicket	New Ticket [201502203549000024] created (Q=Misc,P=3 normal,S=open)	-	Admin OTRS	02/20/2015 09:59:33
ServiceUpdate	Updated Service to Service 1 (ID=1)	-	Admin OTRS	02/20/2015 09:59:33
SLAUpdate	Updated SLA to SLA 1 (ID=1)	-	Admin OTRS	02/20/2015 09:59:33
CustomerUpdate	Updated: CustomerID=customer-1,CustomerUser=customer-1;	-	Admin OTRS	02/20/2015 09:59:33
PhoneCallCustomer	Customer called us	Zoom view	Admin OTRS	02/20/2015 09:59:33
EscalationStart	Triggered "Start" for escalation "First Response (4)" with base time "2015-02-20 09:59:33".	-	Admin OTRS	02/20/2015 09:59:33

Fig. 10: Escalation Debug Mode – Select Escalation Type

After clicking on a escalation type you will get a detailed information about the parameters, attribute values and conditions.

History of Ticket#: 201502203549000024 - test
[Close](#)

OTRSAdvancedEscalations - EscalationType Debug Information

History Content					
ACTION	COMMENT	ZOOM	USER	CREATETIME	DEBUG FIRST RESPONSE
NewTicket	New Ticket [201502203549000024] created (Q=Misc,P=3 normal,S=open)	-	Admin OTRS	02/20/2015 09:59:33	START
ServiceUpdate	Updated Service to Service 1 (ID=1)	-	Admin OTRS	02/20/2015 09:59:33	FixedFilters:
SLAUpdate	Updated SLA to SLA 1 (ID=1)	-	Admin OTRS	02/20/2015 09:59:33	FILTER
CustomerUpdate	Updated. CustomerID=customer-1,CustomerUser=customer-1;	-	Admin OTRS	02/20/2015 09:59:33	AgentMessagePresent
PhoneCallCustomer	Customer called us.	Zoom view	Admin OTRS	02/20/2015 09:59:33	AgentMessageSent
EscalationStart	Triggered "Start" for escalation "First Response (4)" with base time "2015-02-20 09:59:33".	-	Admin OTRS	02/20/2015 09:59:33	CreatedByCustomer
EscalationStart	Triggered "Start" for escalation "Update Time (5)" with base time "2015-02-20 09:59:33".	-	Admin OTRS	02/20/2015 09:59:34	CustomerMessagePresent
EscalationStart	Triggered "Start" for escalation "Solution Time (6)" with base time "2015-02-20 09:59:33".	-	Admin OTRS	02/20/2015 09:59:34	CustomerMessageSent
					Attributes:
					ATTR CURR PREV WAS
					Priority 3 normal
					Queue Misc
					State open
					StateType open
					TicketType Unclassified
					Attribute Sets:
					SET ATTRIBUTES
					1 AgentMessagePresent, Is_StateType, WasNever_StateType

Fig. 11: Escalation Debug Mode – Debug Snapshot of Escalation Type

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 successfully (success = green, red = no change of the state). Basically for each snapshot it is possible that the escalation will calculate multiple state if it do not match to the first.

The order of the calculated states is based on the running, suspend 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.

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.

- AgentMessagePresent: A message from an agent is present.
- AgentMessageSent: An agent sent a message.
- CustomerMessagePresent: A message from a customer is present.
- CustomerMessageSent: A customer sent a message.

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:

- **ATTR:** This column contains the ticket attribute.
- **CURR:** This column contains the current value of the snapshot. This value will be used for the calculation of *equals now*, *doesn't equals now* and *change to* conditions.
- **PREV:** This column contains the value of the previous snapshot. This value will be used for the calculation of *equals previous*, *doesn't equals previous* and *change 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 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.

There are 2 columns:

- **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.
- **ATTRIBUTES:** This column shows the matched conditions of an attribute set. At this point it will use internal names for matching the attributes. Here are some examples:
 - **AgentMessagePresent:** A message from an agent is present.
 - **AgentMessageSent:** An agent sent a message.
 - **CustomerMessagePresent:** A message from a customer is present.
 - **CustomerMessageSent:** 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 matched not if red.

4.2.2 Reports

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

Statistics

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.

4.3 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

Note: Not compatible with the following feature add-on:

- *Ticket Allocation*
-

5.1 Administrator Interface

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

5.1.1 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.

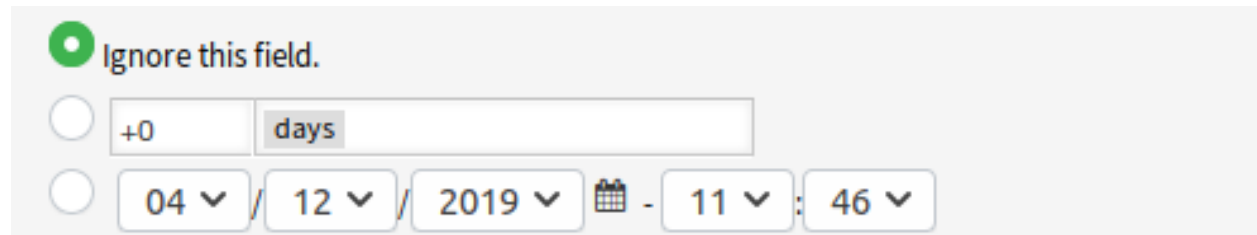


Fig. 1: Date Field in Generic Agent Screen

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

5.2 Agent Interface

This package has no agent interface.

5.3 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

6.1 Administrator Interface

This package has no administrator interface.

6.1.1 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.

6.2 Agent Interface

This package has no agent interface.

6.2.1 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.

6.3 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 FAQ entries. This can refer to the FAQ entry itself or to a specific task that is related to the FAQ 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 FAQ documentation, everyone knows the steps of the task.

Example An employee creates a FAQ entry 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 FAQ or to perform the activity described in the FAQ entry. For new employees who do not have experience, the FAQ makes it possible for them to perform the activity too, because the steps are clearly described in the documentation.

Benefits

- Keep FAQs 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

7.1 Administrator Interface

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

7.1.1 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 FAQ items.

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: *[FAQ field index] → [FAQ 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 FAQ 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.

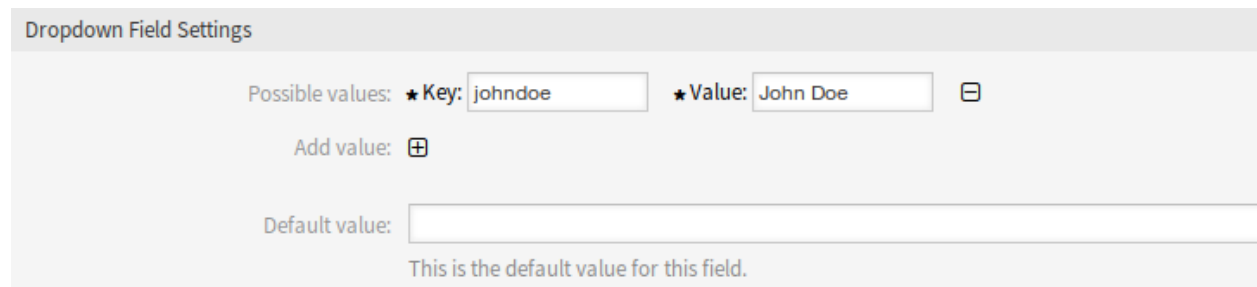


Fig. 2: Edit Dynamic Field Value

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`

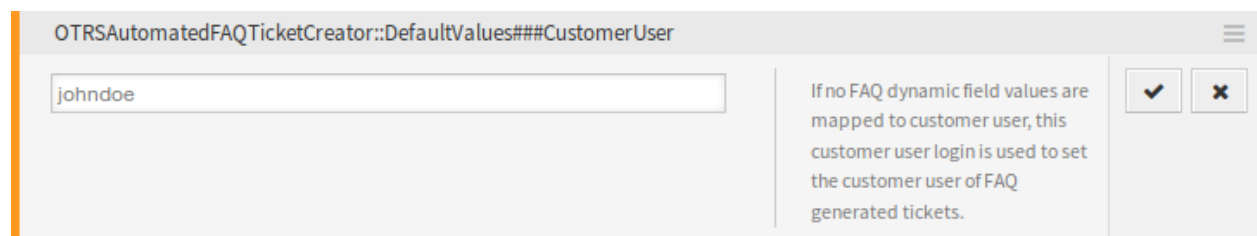


Fig. 3: System Configuration – Default Value

To copy a field value of a FAQ 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 FAQ field as value.

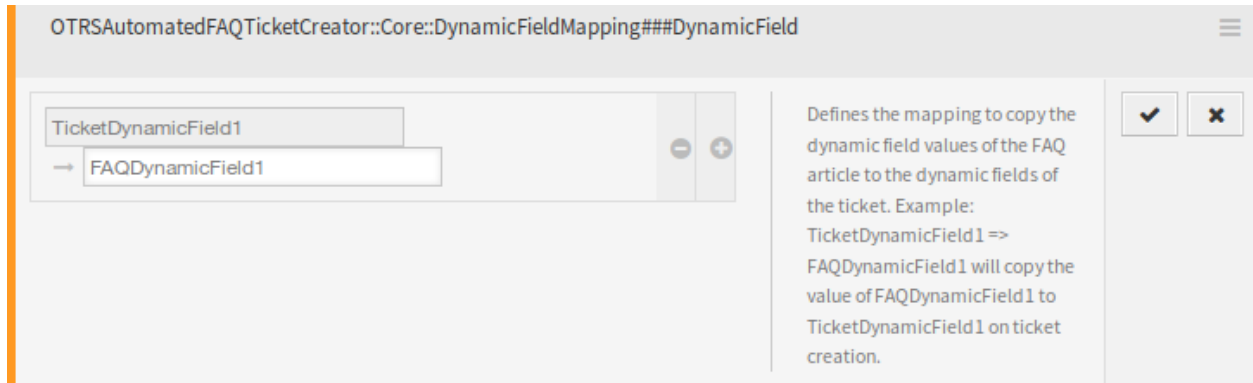


Fig. 4: System Configuration – Copy Value

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*

7.2 Agent Interface

This package has no dedicated agent interface, only the *New* FAQ item screen is updated with new dynamic fields.

7.2.1 FAQ

After installation of the package some new dynamic field are added to the *New* screen of *FAQ* menu.

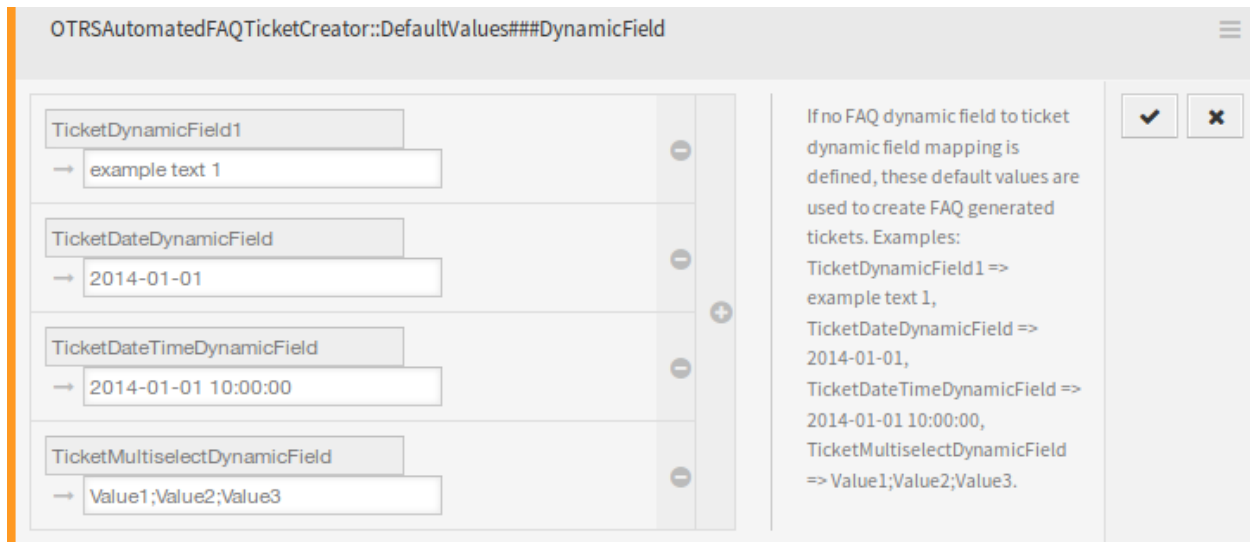


Fig. 5: System Configuration – Default Field Value

New

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 FAQ 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 `FAQ::Frontend::AgentFAQAdd###DynamicField` and/or `FAQ::Frontend::AgentFAQEdit###DynamicField`.
3. Update their values from *0* to *1* as any other dynamic field.

Use Case Example

The following example assumes that all FAQ dynamic fields that set ticket values are activated for the FAQ add screen and the configuration of each dynamic field is set with valid values for the system.

To create a FAQ item which will be triggered for the ticket creation:

1. Create a new FAQ item.
2. Set *2019-04-15 10:00:00* for *Ticket create start time*.
3. Set *2020-04-15 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

Add FAQ Article

All fields marked with an asterisk (*) are mandatory.

★ Title:

Keywords:

★ Category:

State:

Validity:

Language:

Ticket create start time: / / - :

Ticket create end time: / / - :

Ticket create frequency:

Ticket create repeat times:

Ticket create repeat on days:

Fig. 6: Create New FAQ Article

Now every Monday and Friday in the range of 2019-2020 years the OTRS daemon cron job will create a ticket for this FAQ item.

7.3 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

8.1 Administrator Interface

This package implements a scheduled automatic creation of process tickets.

8.1.1 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.

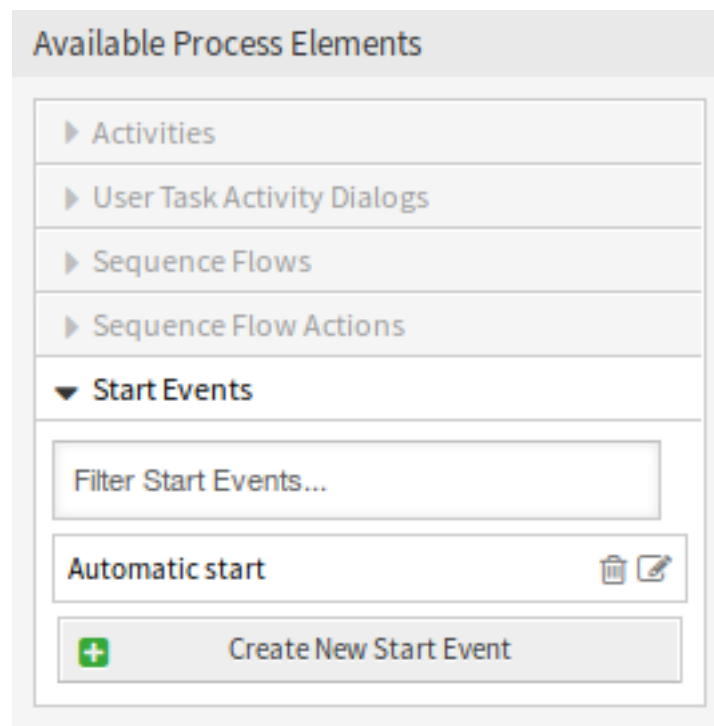


Fig. 1: Start Events

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

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
00 <input style="width: 10px;" type="text"/>	00 <input style="width: 10px;" type="text"/>	Sun <input style="width: 10px;" type="checkbox"/> Mon <input style="width: 10px;" type="checkbox"/> and 5 more... <input style="width: 10px;" type="button" value="⌵"/>

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.

8.2 Agent Interface

This package has no agent interface.

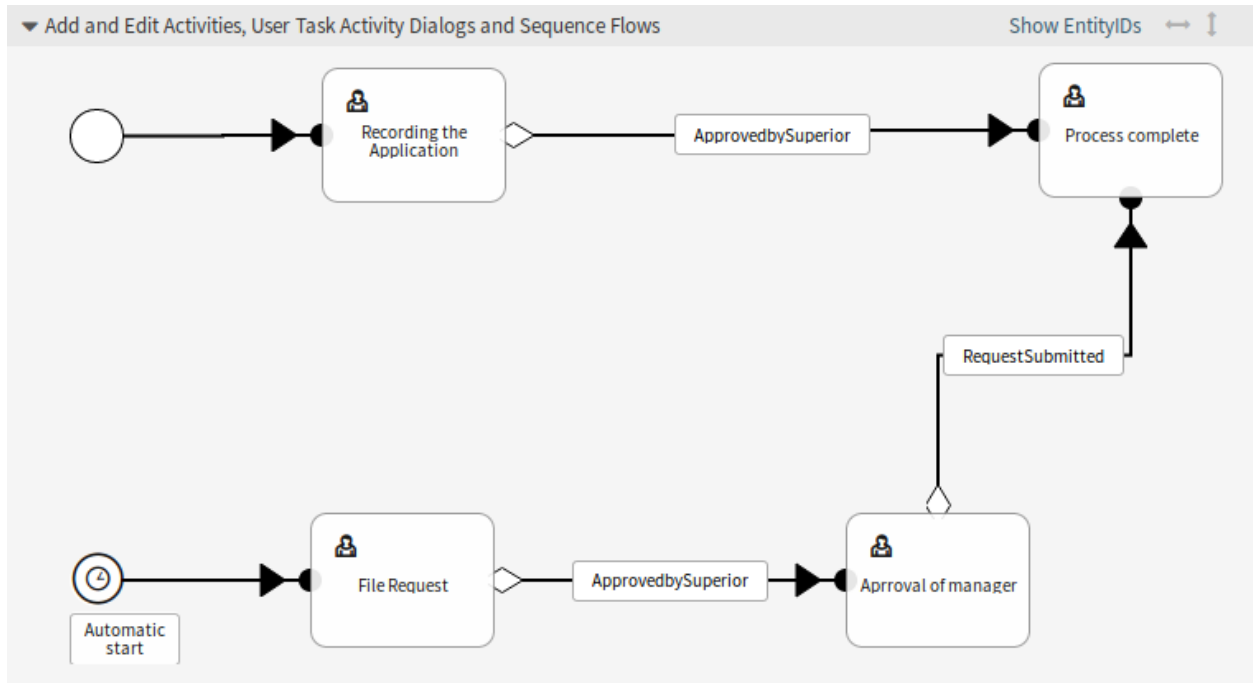


Fig. 4: Start Event on Canvas

8.3 External Interface

This package has no external interface.

Brute Force Attack Protection

Access to your customers' data must be protected at all times, and new defense mechanisms are being created all the time to help with this. A drastic and effective method is to simply block access once too many incorrect logins have been tried. This method can be incorporated into **OTRS**. Whether over the Internet or via VPN, your customers' access to the Self Service Center is now even better protected.

Benefits

- Increases the security of **OTRS**.
- Protects from misuse of your customer accounts.

Target Groups

- Customer service
- E-commerce
- Government
- IT service
- Service providers

Available in Service Package

- SILVER

Package Name in OTRS Package Manager

- OTRSBruteForceAttackProtection

9.1 Administrator Interface

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

9.1.1 Users, Groups & Roles

After installation of the package a new module *Customer Users (Locked)* will be available in the *Users, Groups & Roles* group.

Customer Users (Locked)

Use this screen to review and reset the customer user logins. The locked customer user login management screen is available in the *Customer Users (Locked)* module of the *Users, Groups & Roles* group.

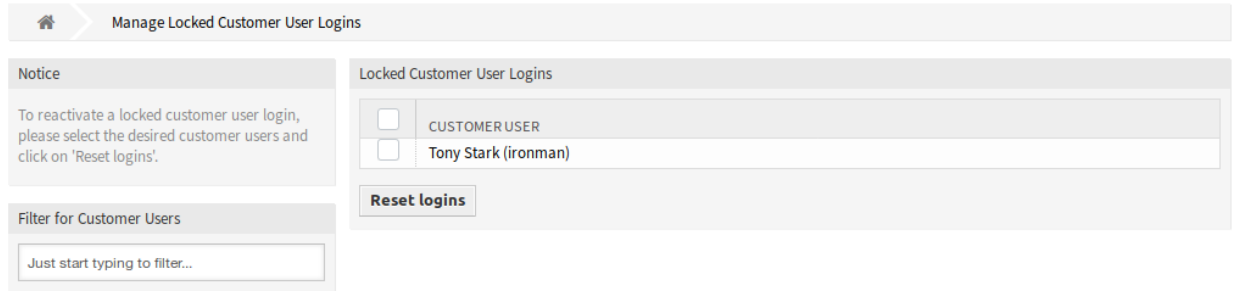


Fig. 1: Locked Customer User Logins Management Screen

Manage Locked Customer User Logins

To reset a customer user login:

1. Select a customer user from the list.
2. Click on the *Reset logins* button.

Select the checkbox in the header of the table to reset all customer user logins.

Note: If several customer users are locked, use the filter box to find a particular customer user by just typing the name to filter.

9.1.2 Administration

After installation of the package a new system configuration option will be available to restrict the passwords in the external interface.

System Configuration

The customer user passwords can be restricted in the external interface to avoid using of weak passwords.

To activate these features:

1. Go to *System Configuration* screen.
2. Select *OTRSBruteForceAttackProtection* in the *Navigation* widget.
3. Navigate to *Frontend* → *External* → *View* → *Preferences* in the navigation tree.

4. Modify the settings.
5. Deploy the modified settings.

Fig. 2: CustomerPersonalPreference###Password Setting

PasswordMaxLoginFailed Allows to lock a customer user to forbid the login if maximum failed logins reached. Setting this option to 0 disables the feature, setting the option to 3 means that only three login attempts possible.

The OTRS log file contains an error message related to the locked customer user account and the forbidden login. This error message can be used by third-party tools like [fail2ban](#).

```
[Mon Oct 15 15:57:03
→2018] [Error] [Kernel::System::CustomerAuth::Auth][259] CustomerUser:
Login for 'customer-1' is forbidden due to too many failed logins!
→ (REMOTE_ADDR: 127.0.0.1)
```

PasswordMin2Characters Defines if the password needs to contain at least 2 letter characters by setting the option to 1.

PasswordMin2Lower2UpperCharacters Defines if at least 2 lowercase and 2 uppercase letter characters are needed by setting the option to 1.

PasswordMinSize Defines the minimum number of characters in passwords.

PasswordNeedDigit Defines if the password needs to contain at least 1 digit by setting the option to 1.

PasswordRegExp Allows to match passwords against a regular expression.

9.2 Agent Interface

This package has no agent interface.

9.3 External Interface

The login in the external interface counts the failed login attempts. If the configurable count of failed login attempts is reached, the customer user account is locked using the customer preferences within the OTRS database.

The login is not possible and the customer user will see the message: *Login failed! Your user name or password entered incorrectly.*

This mechanism works for customer users which are stored in the OTRS database itself or in a tethered LDAP server.

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

10.1 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 *Manage Teams* and *Manage Team Agents* menu item of the *Calendar* menu in the agent interface. It is also possible to send out ICS files in appointment notifications.

10.1.1 Administration

This package extends the *Calendar Management* screen with a new section to set default values and add dynamic fields to appointments.

Calendars

Use this screen to add default values to appointments. The calendar management screen is available in the *Calendars* module of the *Administration* group in the administrator interface or from the *Manage Calendars* menu item of the *Calendar* menu in the agent interface.

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.

10.1.2 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.

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.

▼ Appointment Filter

Calendar:

Title:

Location:

Team:

Resource:

Dynamic Fields

Equipments:

Fig. 2: Appointment Filter Section

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 *Timeline Custom* 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.

10.1.3 Processes & Automation

This package adds support for dynamic fields to appointments.

Note: The usage of *Dynamic Field Database* feature is not supported.

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*.

Dynamic fields are available in the linked appointments widget in the *Ticket Zoom* screen as well.

10.2 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 dynamic list statistics.

Note: The agent have to be the member of `admin` group to access *Manage Calendars*, *Manage Teams* and *Manage Team Agents* screens.

10.2.1 Calendar

After installation of the package, some new menu items will be available in the *Calendar* menu.

Resource Overview

While it is possible to assign resources to an appointment in any calendar overview, this screen is designed exclusively for allocating resources to appointments. The resource overview screen is available in the *Resource Overview* menu item of the *Calendar* menu.

This screen looks very much like a regular calendar overview. It has a filter for teams on top and a list of agents for that team (available only in time-line views). By choosing a team from the top filter, you will be presented with all appointments assigned to it. A new button *Timeline Custom* is also added as a new customizable view.

Management screens for the *Resource Overview* screen:

- Click on the *Manage Calendars* button to set up some calendars.
- Click on the *Manage Teams* button to create teams.
- Click on the *Manage Team Agents* button to assign agents to teams.
- Click on the *Add Appointment* button to create appointment in the configured calendars.

Note: The buttons are available only, if calendars, teams and team agent are configured properly.

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

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 screen. An appointment with multiple assigned agents will be displayed in each row of assigned agents. Dragging one instance of this appointment will move all them in the same fashion.

By editing settings of the resource overview via gear icon in top right corner, you can choose which team agents are displayed within the overview and you can customize the *Timeline Custom* view.

Resource Overview

Actions

- Manage Calendars
- Manage Teams
- Manage Team Agents
- Add Appointment

Calendars

Just start typing to filter...

	NAME	URL
<input checked="" type="checkbox"/>	Company Events	
<input checked="" type="checkbox"/>	Meetings	

Calendar

Team: My Team

Month Week Day Timeline Month Timeline Week Timeline Day Timeline Custom

Jump Today < >

25 Nov - 1 Dec 2019 #48

RESOURCES	MON, 25 NOV											TUE, 26 NOV						
NAME	00	02	04	06	08	10	12	14	16	18	20	22	00	02	04	06	08	10
John Doe						Test appointment												
Unassigned																		

Fig. 3: Resource Overview Screen

There is an other icon in the top right corner labeled with double arrow. Clicking on this icon makes the current view full screen and hides the left sidebar. If you click on the full screen icon again, the view is restored and the left sidebar will be visible. The full width mode is stored in the user preferences, so the full screen mode is remembered even after reloading the overview screens.

The *Timeline Custom* view with the extended settings dialog and the full screen mode is available also in the *Calendar Overview* screen.

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

Timeline Custom Settings

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

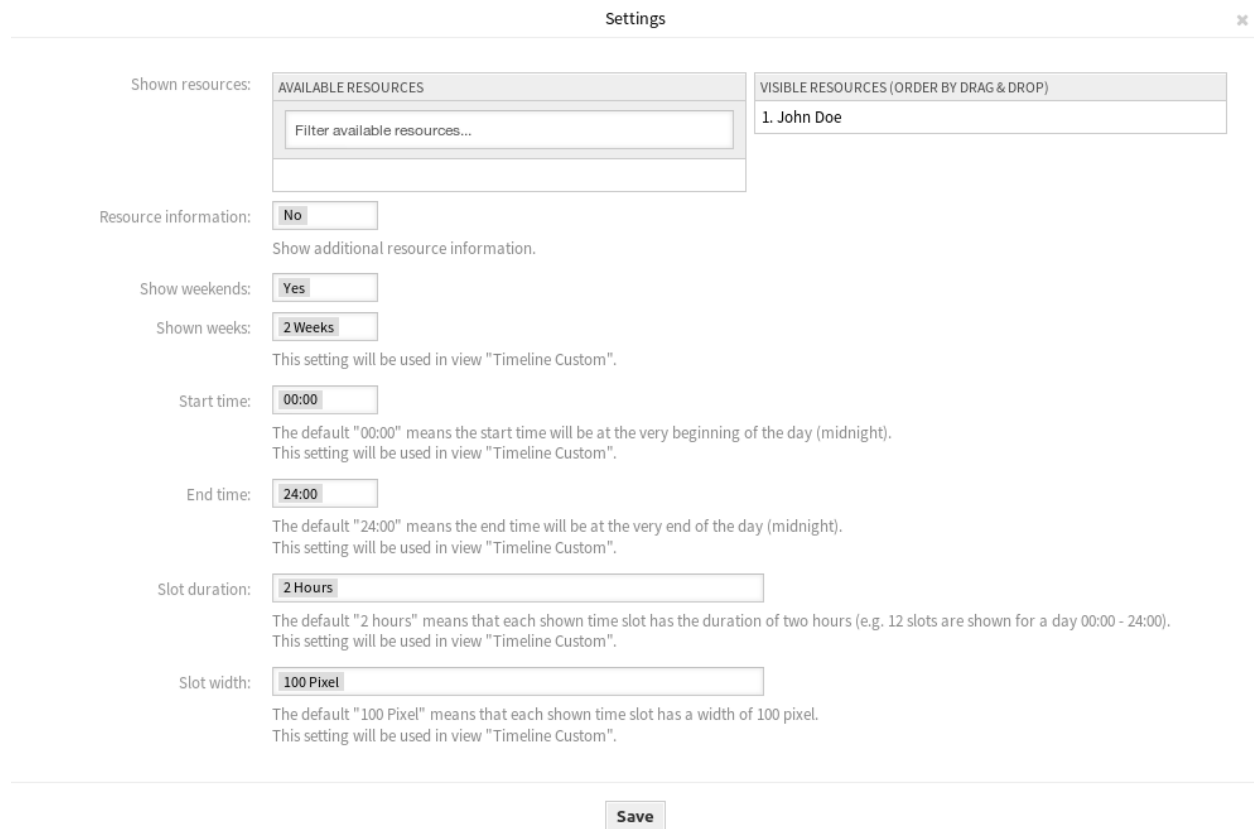


Fig. 4: Calendar Settings Dialog

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

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

Resource information Show additional resource information next to the agent name in the *Resource Overview* screen. 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.

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

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

Start time Select the start time for the *Timeline Custom* 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 *Timeline Custom* 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 *Timeline Custom* 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 *Timeline Custom* view. The default *100 Pixel* means that each shown time slot has a width of 100 pixel.

In *Timeline Custom* 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 *Resource Overview* screens.
2. Click on the gear icon next to a calendar in the *Calendars* 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.

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

Manage Calendars

Note: You have to be the member of `admin` group to access this screen.

This menu item opens the *Calendars* management screen of the administrator interface.

Appointment ✕

Basic information

★ Title:

Description:

Location:

★ Calendar:

Resource

Team:

Agent:

Date/Time

Start date: / / - :

End date: / / - :

Save

Fig. 5: Add Appointment Dialog

Settings for subscription of calendar Company Events ✕

Teams:

You can specify teams which will be used to filter out appointments.

Resources:

You can specify resources which will be used to filter out appointments.

Fig. 6: Calendar Subscription Settings Dialog

Manage Teams

Note: You have to be the member of `admin` group to access this screen.

Use this screen to manage teams. The team management screen is available in the *Manage Teams* menu item of the *Calendar* menu. Additionally, it is also accessible from the *Resource Overview* and *Manage Team Agents* screens.

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

To manage team agents:

1. Click on the *Manage Team Agents* button.

Manage Teams

Actions

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"/>

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

Fig. 7: Manage Teams Screen

Add Team

Team

★ Name:

★ Permission group:

Comment:

★ Validity:

or Cancel

Fig. 8: Add Team Screen

Edit Team

Team

★ Name:

★ Permission group:

Comment:

★ Validity:

or or Cancel

Fig. 9: Edit Team Screen

2. Assign agents to teams and vice versa in the *Manage Team Agents* 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.


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)	

Fig. 10: 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.

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

Fig. 11: Team Import Widget

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.

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.

Manage Team Agents

Note: You have to be the member of `admin` group to access this screen.

Once you have set up a team in *Manage Teams* 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 *Manage Team Agents* menu item of the *Calendar* menu. Additionally, it is also accessible from the *Resource Overview* and *Manage Teams* screens.

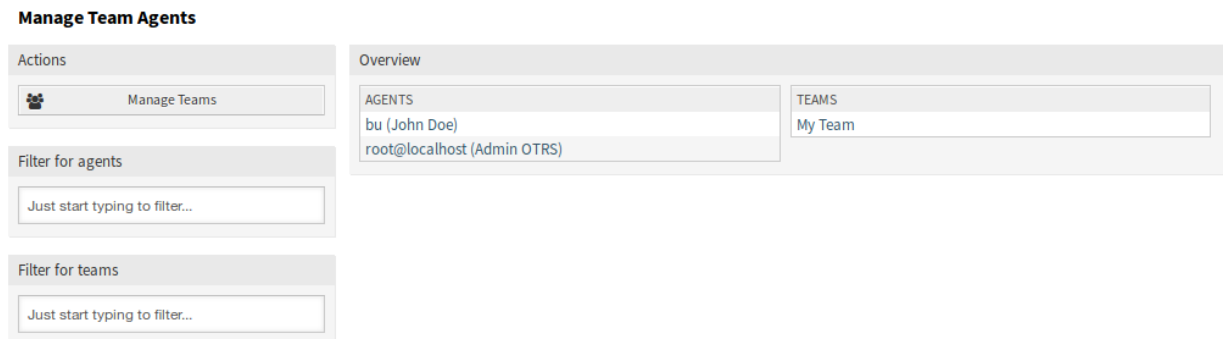


Fig. 12: Manage Team Agents Screen

Manage Agents Teams Relations

To assign some teams to an agent:

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

To assign some agents to a team:

Change Team Relations for Agent Doe John (bu)

TEAM	<input checked="" type="checkbox"/> ACTIVE
My Team	<input checked="" type="checkbox"/>

Save or **Save and finish** or Cancel

Fig. 13: Change Team Relations for Agent

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

AGENT	<input type="checkbox"/> ACTIVE
Doe John (bu)	<input checked="" type="checkbox"/>
OTRS Admin (root@localhost)	<input type="checkbox"/>

Save or **Save and finish** or Cancel

Fig. 14: 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.

10.2.2 Reports

After installation of the package a new object type `AppointmentList` will be available to create dynamic list statistics.

Statistics

After installation of the package you can set up new statistics using the `AppointmentList` statistic backend 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.

10.3 External Interface

This package has no external interface.

Categories for Text Modules

The standard **OTRS** help desk allows your service agents to quickly respond to normal or recurring service requests by leveraging pre-defined text modules to answer the tickets. These text modules are displayed in a single unsorted list, which may cause difficulty when there are a large number of pre-defined text modules. The feature add-on allows you to group certain text modules by category and browse through these categories by expanding or minimizing with a single click in Windows Explorer.

For example, a typical recurring question is the request to send out a user guide for one of your specific products. Today, you would probably browse down the long list of your pre-defined text modules and, once you finally have identified the right one with the user guide attached, you would send it out.

With this feature add-on, you consolidate all text modules related to your products' user guides in a separate category like *User Guides*. You can find it quickly, saving a lot of time trying to retrieve this information. This reduces your response time.

Benefits

- By building template categories recurrent requests can be answered even faster.
- Facilitates job training of new agents.

Target Groups

- IT service management
- Call centers
- Service providers
- Any company working with SLAs

Available in Service Package

- GOLD

Package Name in OTRS Package Manager

- OTRSCategoriesForTextModules

Note: Not compatible with the following feature add-ons:

- *Advanced Editor*
 - *Custom Contact Fields*
 - *Hide/Show Dynamic Fields*
 - *Ticket Forms*
-

11.1 Administrator Interface

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

11.1.1 Ticket Settings

After installation of the package two new modules will be available in the administrator interface and the template management screen is extended with new fields.

Categories

Use this screen to add categories to the system. The category management screen is available in the *Categories* module of the *Ticket Settings* group.

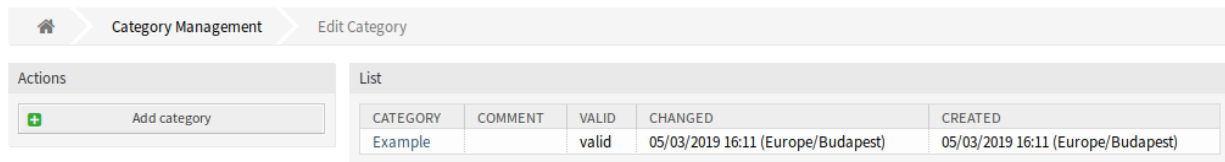


Fig. 1: Category Management Screen

Manage Categories

To add a category:

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

Warning: Categories 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 category:

1. Click on a category in the list of categories.
2. Modify the fields.

Fig. 2: Add Category Screen

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

Fig. 3: Edit Category Screen

Category Settings

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

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

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

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.

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.

Templates

The *Template Management* screen is extended with new fields. A template can now have to, cc and destination queue properties.

Add Template

★ Type:

★ Name:

Subject:

Subject method:

Specify if the current article subject should be Combined, Kept or Overwritten with the subject from the template.

Subject:

Template:

B I U S | | | | | | | | | | | |

Format | Font | Size | | | | |

To:

Cc:

Queue:

Attachments:

★ Validity:

Comment:

or

Fig. 4: Add Template Screen

New Fields

To The person, to whom the ticket is created for. This will be the *To* field of the email.

Cc Other addresses as carbon copy of the email, if needed.

Bcc Other addresses as blind carbon copy of the email, if needed.

Templates Categories

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



Fig. 5: Manage Template-Category Relations Screen

Manage Templates Categories Relations

To assign a template to a category:

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

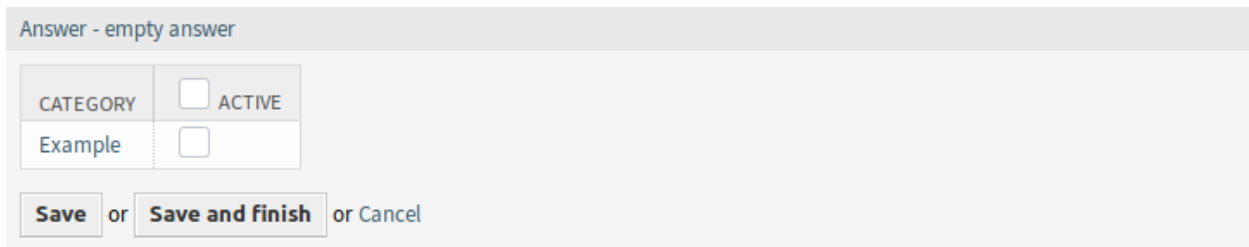


Fig. 6: Change category Relations for Template

To assign a category to a template:

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

Example

TEMPLATE	<input type="checkbox"/> ACTIVE
Answer - empty answer	<input type="checkbox"/>
Answer - test answer	<input type="checkbox"/>

or or

Fig. 7: Change Template Relations for Category

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

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

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

11.2 Agent Interface

This package includes the functionality to show categories for standard templates as a folder structure.

11.2.1 Tickets

This package implements a category folder structure for standard templates. This folder structure will be shown in the *Ticket Zoom* screen if clicking on the *Reply* button in the article menu.

New Phone Ticket

Use this screen to create tickets based on phone calls. The phone ticket screen is available in the *New Phone Ticket* menu item of the *Tickets* menu.

New Widget

A new widget named *Templates by category* is added to the left side of this screen, where the templates can be selected in a category tree.

Ticket Zoom

After installation of the package some new features are added to the *Ticket Zoom* screen.

Create New Phone Ticket

All fields marked with an asterisk (*) are mandatory.

★ Customer user:

Customer ID:

★ To queue:

Service:

Service Level Agreement:

Owner:

★ Subject:

Options: [Customer user] [FAQ]

Text Template:

Setting a template will overwrite any text or attachment.

Customer Information

none

Templates by category

Show view:

Fig. 8: New Phone Ticket Screen

Article Zoom

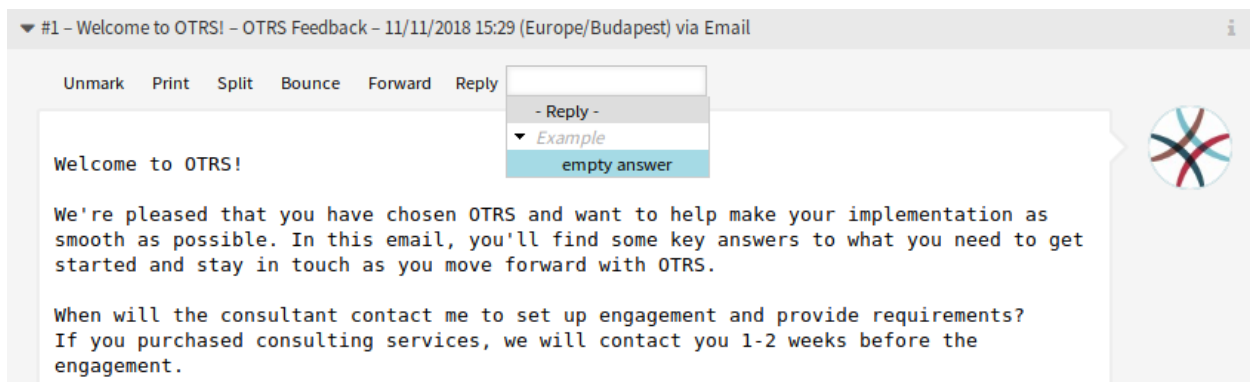


Fig. 9: Article Zoom Widget

Reply Use this menu item to compose an answer to the article via email. Reply is created from a response template set in *Templates* module of the administrator interface.

The menu item is extended with a tree view, so templates can be selected based on the defined categories.

Article Notification

This package also adds the functionality to inform agents about updates on tickets they are currently working on.

If an agent opened the *Ticket Zoom* and started working on a ticket (adding an answer, a note, etc.), a pop-up is opened for this work. If the ticket is updated by another agent at this time, the agent will be informed by a dialog box in the pop-up. He has the possibility to close the pop-up without saving to view the updates on the tickets or to continue working within the pop-up without updating the ticket.

This functionality can be also activated in the main ticket zoom window by activating a setting in the system configuration.

11.3 External Interface

This package has no external interface.

Change Write Protection

This feature add-on makes it possible to hide specific changes or work orders in a particular status, thus preventing further changes.

The feature add-on is ideal for use in IT service management. For example, the authorization for a change normally comes after its description. After authorization is given, the description of the change can no longer be changed retroactively in the change builder. In addition, the change is then no longer shown in the *Change Zoom* screen.

With system configuration you can define:

- The status of a change/work order: enables the creation of a list of statuses that must take place for write protection to be activated.
- Actions: enables the creation of a list of actions that are no longer shown/no longer possible once a predefined status is activated.

This feature add-on will take you to the next level of transparent work and careful change management.

Benefits

- Increases transparency and leads to more care and diligence in the change process.
- Optimizes the general transparency of your system.

Target Groups

- IT service management
- All customers that use the OTRS::ITSM feature

Available in Service Package

- GOLD

Package Name in OTRS Package Manager

- OTRSChangeWriteProtection

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

12.1 Administrator Interface

This package has no administrator interface.

12.2 Agent Interface

This package has no agent interface, but the menu items of the change menu and work order menu can be restricted.

12.2.1 ITSM Changes

After installation of the package, it is possible to list states and forbidden actions, which are no longer visible and usable, once a change or a work order is in a defined state.

Overview

This feature makes it possible to *write protect* a change or a work order, which means that certain actions which are normally visible in the *Change Zoom* or the *Work Order Zoom* screens, are no longer visible and usable, once a change or a work order is in a defined state.

See also:

The list of states and forbidden actions can be individually configured for changes and work orders in the following system configuration settings:

- `ITSMChange::WriteProtection###Actions`
- `ITSMChange::WriteProtection###ChangeStates`
- `ITSMWorkOrder::WriteProtection###Actions`
- `ITSMWorkOrder::WriteProtection###WorkOrderStates`

If a change is write protected, all included work orders will be write protected as well. This feature can be disabled in the system configuration.

If you want to hide some work order or change actions from the zoom screen if the work order or change has reached a certain state, you need to define the states and actions in the system configurations.

Then look at the the zoom screen of a work order or a change which has not yet reached one of the defined states and notice that still all actions are visible in the zoom screen.

Now change the state of the work order or change to a state that was defined in the write protection settings, and notice that the actions listed in the system configuration settings are no longer visible. These actions can also not be called directly, if you would have bookmarked the link to that action.

12.3 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 CI-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 CI 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 CI 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 CIs to services and SLAs.
- Enables the provision of CI-oriented services with numerous CI attributes.
- Greater flexibility and transparent linking of CI attributes to ticket dynamic fields.
- Optimized visualization of CI attributes with multiple values in a drop-down in the ticket.
- CI attributes are readable and searchable.

Target Groups

- Companies that offer services for different devices/products (CIs)
- 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 *ITSM Configuration Management* feature.

13.1 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.

13.1.1 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 *Config 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.

13.1.2 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.

13.2 Agent Interface

This package has no agent interface.

13.3 External Interface

This package has no external interface.

CI Custom Search

This feature add-on makes it possible to use an additional user-defined search for configuration items. This adds the new menu section *Search (Custom)* to the main *CMDB* menu.

Regardless of the assigned CI class, you can now search for a specific owner and view all CIs used by the owner, such as printer, phone or email client.

You can also search for CIs in a specific CI class, as well as for a specific deployment or incident state. Moreover, you can configure the name of the newly displayed menu section and add further CI attributes you wish to use in the search mask.

Benefits

- Faster and more efficient search for configuration items related to a specific customer user.
- Simplified change management in case of new employees.
- Better customer service when a wide range of products is used by a customer.

Target Groups

- Internal IT departments
- External IT service providers
- Customer support for a wide range of products

Available in Service Package

- GOLD

Package Name in OTRS Package Manager

- OTRSCICustomSearch

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

14.1 Administrator Interface

This package has no administrator interface.

14.1.1 Add Search Attribute

It is possible to add additional search attribute to the search dialog.

To add a search attribute:

1. Go to *System Configuration* screen.
2. Select *OTRSCICustomSearch* in the *Navigation* widget.
3. Navigate to *Frontend* → *Agent* → *View* → *ConfigItemSearch* in the navigation tree.
4. Search for setting `ITSMConfigItem::CustomSearchXMLFields` and add new search attribute. The key is the value from the `Key` entry of the configuration item.

If the field is set to *0 - Disabled*, the search attribute will not be displayed in the search dialog.

14.2 Agent Interface

This module offers the possibility to use an additional search for configuration items. This search functionality is used to search for configuration items across several configuration item classes, which all belong to a specific owner.

14.2.1 CMDB

After installation of the package a new menu item will be available in the *CMDB* menu in the agent interface.

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

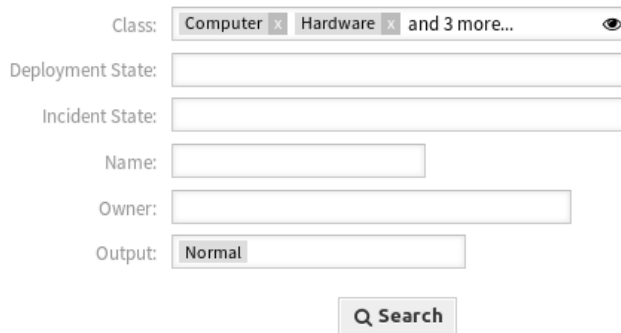
The menu items that were added by this package will be visible after you log-in to the system again.


Search (Custom)

Use this screen to search for configuration items with custom search.

To search for configuration items:

1. Click on the *Search (Custom)* menu item in the *CMDB* menu.
2. Select entries from the drop-down fields and add search terms in the text fields.
3. Select the output format.
4. Click on the *Search* button.
5. See the search result.

Search Config Item:

Class: Computer x Hardware x and 3 more... 

Deployment State:

Incident State:

Name:

Owner:

Output: Normal

Fig. 1: Search Config Item Dialog

The search can be limited to specific configuration item classes. All classes are selected by default. Additionally, it is possible to limit the search by selecting the deployment and/or incident state.

The main use case is to search for configuration item owners across several configuration item classes. For example it is possible to find all configuration items of a specific owner during a change of an employee.

See also:

Additional search fields can be added in the system configuration in the [Administrator Interface](#).

14.3 External Interface

This package has no external interface.

CI References

With this feature add-on it is possible to enhance the CIs 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 CIs with each other when there is a dependency relationship between them or to store information regarding linked services and users directly in the CI, 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 *Config Item*. Then, it is necessary to add the visualization configuration as code for the desired CI class. If, for example, a new CI 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 CIs using autocomplete can then be carried out, and these can be added. If the new CI is updated, an automatic linking to the referenced CI is carried out.

Benefits

- Enhancement of CIs through additional input fields and additional, system-referenced data.
- Linking of CIs with each other and with other data in **OTRS**.
- Quicker autocomplete search for values referenced in CI fields.
- Clear visualization of complex dependencies between CIs and other CIs, 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 *ITSM Configuration Management* and *Import/Export* features.

15.1 Administrator Interface

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

15.1.1 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 *Config 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. 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
```

(continues on next page)

(continued from previous page)

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. 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. 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`.

15.2 Agent Interface

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

15.2.1 CMDB

Select the *New* menu item from the *CMDB* menu in order to see the new input fields for configuration items.

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

The menu items that were added by this package will be visible after you log-in to the system again.

New

Use this screen 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.

Hard Disk:

Capacity:

Hard Disk:

FQDN:

★ Network Adapter:

★ IP over DHCP:

IP Address:

Network Adapter:

Graphic Adapter:

Other Equipment:

Warranty Expiration Date: / /

Install Date:

Note:

Test CI:

Test Service:

Test User:

Fig. 1: Add Configuration Item To Class Computer

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

See also:

The fields in the *Add 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.

15.3 External Interface

This package has no external interface.

CI in Customer Frontend

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

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

On mail-in this feature add-on automatically links affected CIs 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 CIs of the requester's organization or department. This eases the selection of affected CIs a lot, especially if you have many CIs in your CMDB.

Benefits

- Automatic assignment of CIs saves time.
- Customers can assign CIs 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: Not compatible with the following feature add-ons:

- *Hide/Show Dynamic Fields*
 - *Ticket Forms*
-

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

16.1 Administrator Interface

This package has no administrator interface.

16.1.1 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 *Config 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.

16.1.2 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.

16.1.3 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 *Config 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.

16.2 Agent Interface

This package has no agent interface.

16.2.1 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`.

16.2.2 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`.

16.3 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.

16.3.1 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.

16.3.2 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`

Home > Company Configuration Items

Company Configuration Items

Computer

Incident State	Deployment State	ConfigItem#	Name	Class	Last changed
Operational	Production	1058000001	Desktop PC #01	Computer	06/11/2019 14:23:28 (UTC)

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 #01

Configuration Item Details

Vendor	
Model	
Description	
Type	
Customer Company	Stark Industries
Owner	"Tony Stark" <tony@stark-industries.com>
Serial Number	
Operating System	
CPU	
Ram	
Hard Disk	
↳ Capacity	
FQDN	
Network Adapter	NIC
↳ IP over DHCP	↳ Yes
Graphic Adapter	
Warranty Expiration Date	06/05/2019
Note	Desktop computer

Configuration Item Information

Class:
Computer

Number:
1058000001

Name:
Desktop PC #01

Current Deployment State:
Production

Current Incident State:
■ Operational

Created:
06/05/2019 13:09:49 (UTC)

Created:
06/11/2019 14:23:28 (UTC)

Fig. 2: Configuration Item Details Screen

CHAPTER 17

Credit Card Filter

This feature add-on offers options for the responsible handling of sensitive data, like credit card numbers, in your **OTRS** system. With the help of this function, it is possible to hide credit card numbers for new tickets and articles, as well as articles already stored in the system. Additionally, flexibly configurable warning messages for the credit card number can be automatically set and showed.

This mechanism for recognizing credit cards works for cards with 13, 15 or 16 digits. All articles of a ticket in the system will be scanned for valid credit card numbers before storing. Then, the majority of the digits will be encoded before storage in the system.

Benefits

- Improved data protection.
- Less security risks for credit card data.

Target Groups

- Customer service
- Finance
- Sales
- Order management
- Complaint management

Available in Service Package

- SILVER

Package Name in OTRS Package Manager

- OTRSCreditCardFilter

17.1 Administrator Interface

This package has no administrator interface.

17.1.1 Legacy Credit Card Masking

A console command exists to treat already stored credit card numbers in the system. This command will mask any unmasked valid credit card number in the article database table.

To mask the existing credit card numbers:

1. Make sure that setting `OTRSCreditCardFilter::ActiveMaskEnabled` is not enabled.
2. Create one or more tickets with valid credit card information.

```
Issuing Network, Card Number
JCB, 3528988095245935
JCB, 3112606824580636
JCB, 3096030869937728
JCB, 3112437499296450
JCB, 3096010732100407
JCB, 3528461498782367
JCB, 3112892137191440
JCB, 3088814635323630
```

3. Open the *Ticket Zoom* screen to make sure that the credit card numbers are not masked.
4. Execute the following command in the command line to mask the credit card numbers:

```
otrs> /opt/otrs/bin/otrs.Console.pl Maint::Ticket::MaskCreditCard --
→restart yes
```

5. Refresh the *Ticket Zoom* screen. The article the body will shown as:

```
Issuing Network, Card Number
JCB, 352898xxxxxx5935
JCB, 311260xxxxxx0636
JCB, 309603xxxxxx7728
JCB, 311243xxxxxx6450
JCB, 309601xxxxxx0407
JCB, 352846xxxxxx2367
JCB, 311289xxxxxx1440
JCB, 308881xxxxxx3630
```

This script starts masking the credit card numbers on last articles first, since they are the most common used, so the results can be seen faster.

See also:

For more information about the `Maint::Ticket::MaskCreditCard` parameters, execute the following command:

```
otrs> /opt/otrs/bin/otrs.Console.pl Maint::Ticket::MaskCreditCard --help
```

The architecture of this script is designed to avoid affecting the system performance by working in batches and waiting between each batch. The number of processed articles per batch and the wait time between batches can be fine tuned to match system performance.

The script is also designed to remember last processed article and start again from that, allowing to stop the process at a certain time and resume later. There is an override to force starting again from the beginning.

You could specify an end date so only articles until that date will be processed (e.g. if you started automatic masking at a certain date), also it is possible to specify the number of articles to process per run.

17.2 Agent Interface

This package has no agent interface.

17.2.1 Mask Credit Card

This feature is a complete subsystem that allows the following:

- Automatically show a warning message next to a credit card number (not storable).
- Mask credit card numbers for new tickets and articles.
- Mask credit card numbers contained in articles already stored in the system.

The credit card detection mechanism requires credit card numbers of 13, 15 or 16 digits. These credit card numbers should be at least potentially valid numerically, which means they pass the [Luhn algorithm](#) test.

For the current version only a subgroup of all potentially valid credit card number are detected. This is the list of the credit cards numbers that are considered valid:

- Visa 16 digits starting with a 4.
- Visa 13 digits starting with a 4.
- MasterCard 16 digits starting with 51 to 55.
- Discover 16 digits starting with 6011, 6121-29 to 6229-25, 644 to 649 or 65.
- JCB 16 digits starting with 3088, 3096, 3112, 3158, 3337 or 3528 to 3589.
- JCB 15 digits starting with 1800, 2100 or 2131.
- American Express 15 digits starting with 34 or 37.

For successful detection, the digits of these credit card numbers are allowed without separation or with a single separator in groups of digits as 4-4-4-4, 4-4-4-3, 4-4-4-1 or 4-6-5 (the last combination for American Express only). Allowed separators are -, +, /, . or a combination thereof.

Valid credit card numbers that are a subset of a bigger number are not considerate as credit card numbers. This is to avoid false positives, e.g. a serial number that contains a (not intentionally) valid credit card number. Valid credit card numbers should be enclosed by at least one non-numeric character.

17.2.2 Active Credit Card Masking

When this feature is enabled, every article will be scanned for valid credit card numbers before it is saved on the database. In case of any findings in subject or body, all but the first six and the last four digits will be replaced by a configurable masking character.

See also:

The behavior can be changed with the following settings in the system configuration:

- `OTRSCreditCardFilter::ActiveMaskEnabled`
- `OTRSCreditCardFilter::MaskedCharacter`

For example `1234-5678-9012-3456` becomes `1234-56xx-xxxx-3456`.

Warning: This procedure is permanent and irreversible!

To use this feature:

1. Create a ticket with the following article body:

```
Issuing Network, Card Number
JCB 15 digit,180061388939823
JCB 15 digit,180079668437698
JCB 15 digit,180001434886883
JCB 15 digit,180044208063503
JCB 15 digit,180010497338476
JCB 15 digit,210004248524033
JCB 15 digit,210012319871803
JCB 15 digit,180094846333594
JCB 15 digit,210084424984649
JCB 15 digit,210012951351973
JCB 15 digit,210008094074787
JCB 15 digit,210081171733450
```

2. Open the *Ticket Zoom* screen to see the created ticket. The body of the article will shown as:

```
Issuing Network, Card Number
JCB 15 digit,180061xxxxx9823
JCB 15 digit,180079xxxxx7698
JCB 15 digit,180001xxxxx6883
JCB 15 digit,180044xxxxx3503
JCB 15 digit,180010xxxxx8476
JCB 15 digit,210004xxxxx4033
JCB 15 digit,210012xxxxx1803
JCB 15 digit,180094xxxxx3594
JCB 15 digit,210084xxxxx4649
JCB 15 digit,210012xxxxx1973
JCB 15 digit,210008xxxxx4787
JCB 15 digit,210081xxxxx3450
```

3. You can also try with valid credit card numbers in the subject or fetch a mail with valid credit card information using a postmaster account.

17.2.3 Credit Card Warning Message

It is possible to display a warning message next to the credit card number.

To use this feature:

1. Make sure that setting `OTRSCreditCardFilter::ActiveMaskEnabled` is not enabled.
2. Make sure that setting `OTRSCreditCardFilter::WarningTextEnabled` is enabled.

3. Define your custom message in setting `OTRSCreditCardFilter::WarningText` in the system configuration.
4. Create a ticket with the following article body:

```
Issuing Network, Card Number
JCB, 3528988095245935
JCB, 3112606824580636
JCB, 3096030869937728
JCB, 3112437499296450
JCB, 3096010732100407
JCB, 3528461498782367
JCB, 3112892137191440
JCB, 3088814635323630
```

5. Open the *Ticket Zoom* screen. The warning message should appear next to the credit card number.

```
Issuing Network, Card Number
JCB, 3528988095245935 Reminder: You should not store credit card numbers
→in this product!
JCB, 3112606824580636 Reminder: You should not store credit card numbers
→in this product!
JCB, 3096030869937728 Reminder: You should not store credit card numbers
→in this product!
JCB, 3112437499296450 Reminder: You should not store credit card numbers
→in this product!
JCB, 3096010732100407 Reminder: You should not store credit card numbers
→in this product!
JCB, 3528461498782367 Reminder: You should not store credit card numbers
→in this product!
JCB, 3112892137191440 Reminder: You should not store credit card numbers
→in this product!
JCB, 3088814635323630 Reminder: You should not store credit card numbers
→in this product!
```

You can also try with valid credit card numbers in the subject or fetch a mail with valid credit card information using a postmaster account.

17.3 External Interface

This package has no external interface.

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 Information Center*, 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

18.1 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.

18.1.1 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 `Core::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 for 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.

18.1.2 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 style="width: 100%;" 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 style="width: 100%;" 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 style="width: 100%;" 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 style="width: 100%;" 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 style="width: 100%;" 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 style="width: 100%;" 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 ticket view screens.

Add Dynamic Field to External Interface

After creating a dynamic field of type *Customer*, it has to be added to the screens, where it should get displayed.

To add the dynamic field to the external interface:

1. Go to *System Configuration* screen.
2. Select *All settings* in the *Navigation* widget.
3. Navigate to *Frontend* → *External* → *View* → *TicketDetailView* in the navigation tree.
4. Search for setting `ExternalFrontend::TicketDetailView###DynamicField` and click on the *Edit this setting* button.
5. Click on the + button to add the dynamic field. The key is the name of the dynamic field, the value is *1 - Enabled*. This setting is used to display the content of dynamic field in the ticket detail view.
6. Search for setting `ExternalFrontend::TicketDetailView###FollowUpDynamicField` and click on the *Edit this setting* button.
7. Click on the + button to add the dynamic field. The key is the name of the dynamic field, the value is *1 - Enabled*. This setting is used to configure dynamic fields in the answer part of customers detail view.
8. Repeat the steps for other views, if needed.
9. Deploy the modified settings.

If other packages are installed along with this package that override template names for certain modules, manual change of setting `Frontend::Output::FilterElementPost###OTRSCustomContactFieldsJS` is needed. Overridden template names will have to be added to the list of output filter configuration so it can work correctly.

For example, package `OTRSCIsInCustomerFrontend` overrides templates `CustomerTicketMessage` and `CustomerTicketZoom` with `OTRSCIsInCustomerFrontendCustomerTicketMessage` and `OTRSCIsInCustomerFrontendCustomerTicketZoom` respectively. These template names have to be added to the system configuration according to example below, in order for output filter to be activated.

18.2 Agent Interface

This package has no agent interface.

18.2.1 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 (*Ticket Zoom* and *Ticket Detail View*) 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 All* functionality is used in the *Ticket Zoom* screen, 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 `AgentTicketCompose`.

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.

See also:

Add the name of the dynamic field to setting `Ticket::Frontend::AgentTicketSearch###DynamicField`.

Depending on the system configuration it is possible that the field has to be selected from the attributes drop-down list and added via clicking the + symbol.

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.

Additionally it is possible to use wildcards in the *Customer* dynamic field search. For example searching for all tickets that have an *otrs.com* user stored in the dynamic field would look like this: `*@otrs.com`. To search all tickets with a *mr.smith* stored to it the criteria would look like this: `mr.smith@*`. If all tickets with a customer user login containing *otrs* should get searched, the search criteria would look like this: `*otrs*`.

18.3 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 Activity Widget

This feature add-on provides a dashboard widget that allows agents to define a set of customers to automatically query. It displays a table showing the related number of open, locked and escalated tickets. Drill down to specific tickets by simply clicking on one of the numbers shown per customer.

Benefits

- Shows the current distribution of tickets at a single glance.
- Tickets can be selected from the overview using just one click.
- Facilitates ticket prioritization.

Target Groups

- IT service providers
- Finance
- Sales
- And many more

Available in Service Package

- SILVER

Package Name in OTRS Package Manager

- OTRSCustomerActivityWidget

Note: Not compatible with the following feature add-ons:

- *Categories for Text Modules*
-

19.1 Administrator Interface

This package has no administrator interface.

19.2 Agent Interface

This package contains a dashboard widget, which displays detailed information about customers and their tickets.

19.2.1 Dashboard

After installation of the package a new widget will be available on the dashboard.

Dashboard

Every agent can choose the customers he wants to see via configuration options of this dashboard widget. For the selected customers the number of open/unlocked, open/locked and escalated tickets will be shown.

Customer Activity			
CUSTOMER COMPANY	OPEN/UNLOCKED	OPEN/LOCKED	ESCALATED
Stark Industries	4	1	0
	TICKET#	SUBJECT	
■	2019060510000015	Memory needed	
■	2019053010000014	New data is needed	
■	2019042610000024	Escalation	
■	2019042610000015	Escalation	

Fig. 1: Customer Activity Widget

By clicking on the number of tickets a list of tickets will be shown. Click on a ticket to open the it in the *Ticket Zoom* screen.

To turn on and off the widget:

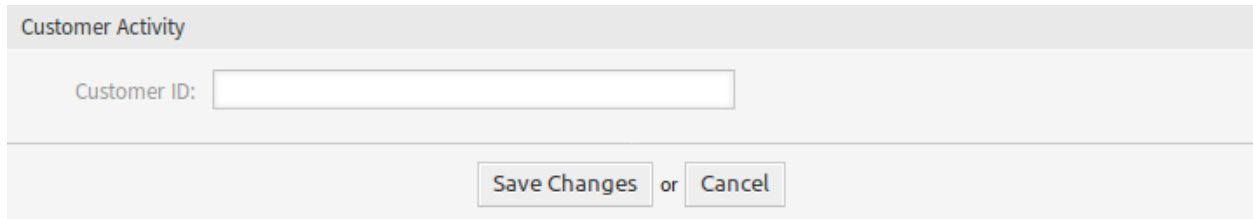
1. Expand the *Settings* widget in the right sidebar by clicking on its header bar.
2. Modify the checked state of the *Customer Activity* entry.
3. Click on the *Save settings* button.

To add customers to the widget:

1. Hover the mouse over the widget header.
2. Click on the gear icon in the top right corner.
3. Enter the customer ID of a customer into the text field.
4. Click on the *Save Changes* button.

After entering and submitting a customer, an additional customer can be entered in the empty company field after re-opening the settings. So customers can be added one after another.

To remove a customer from the widget:



The screenshot shows a web interface for adding a customer. At the top, there is a header bar with the text "Customer Activity". Below this, there is a form with a label "Customer ID:" followed by a text input field. At the bottom of the form, there are two buttons: "Save Changes" and "Cancel", separated by the word "or".

Fig. 2: Customer Activity Widget - Add Customer

1. Hover the mouse over the widget header.
2. Click on the gear icon in the top right corner.
3. Click on the minus icon next to the text field.
4. Click on the *Save Changes* button.

19.3 External Interface

This package has no external interface.

Customer Event Ticket Calendar

With this feature add-on, time periods specified in a ticket are displayed as calendar entries.

This allows appointments to appear in the external interface in such a way that customer users can easily see information about availability or scheduling, such as when they want to find the date of a WebEx meeting.

Benefits

- Appointments are displayed in a calendar format.
- Availability can be seen at a glance.

Target Groups

- Service companies that want to visually represent availability
- Customer service
- ITSM

Available in Service Package

- TITANIUM

Package Name in OTRS Package Manager

- OTRSCustomerEventTicketCalendar

20.1 Administrator Interface

After installation of the package the available configuration settings can be found in the system configuration. This package has no administrator interface.

20.1.1 System Configuration Settings

The *Customer Event Ticket Calendar* widget displays tickets in a calendar, but this widget does not work out-of-the-box.

To display the tickets in this calendar, the following settings need to be set by an **admin user**:

1. Create the following dynamic fields:

Object	Type	Name	Label
Ticket	Date / Time	CustomerTicketCalendarStartTime	Calendar Start Time
Ticket	Date / Time	CustomerTicketCalendarEndTime	Calendar End Time

2. Navigate to *Frontend* → *Agent* → *View* and select a view to add the dynamic field to.

For example add these dynamic fields to *New Phone Ticket* screen and *New Email Ticket* screen to set the dates at ticket creation time, or to *Ticket Free Fields* screen to set the dates for existing ticket in the *Miscellaneous* → *Free Fields* menu item of the *Ticket Zoom* screen.

- Ticket::Frontend::AgentTicketPhone###DynamicField
 - CustomerTicketCreateStartTime → *1 – Enabled*
 - CustomerTicketCreateEndTime → *1 – Enabled*
- Ticket::Frontend::AgentTicketEmail###DynamicField
 - CustomerTicketCreateStartTime → *1 – Enabled*
 - CustomerTicketCreateEndTime → *1 – Enabled*
- Ticket::Frontend::AgentTicketFreeText###DynamicField
 - CustomerTicketCreateStartTime → *1 – Enabled*
 - CustomerTicketCreateEndTime → *1 – Enabled*

3. Add more queues to `CustomerTicketCalendar###Queues` setting (default is *Raw* queue only).

If the dynamic fields contain values for the tickets, the tickets are displayed in the widget.

20.2 Agent Interface

This package has no agent interface.

20.3 External Interface

This package implements the *Events Ticket Calendar* for the external interface. If a ticket has dynamic fields contain date values, the ticket will be displayed in the calendar widget.

See also:

The calendar settings can be set in the system configuration in the [Administrator Interface](#).

When a new ticket is created for the customer user and the values are set in the dynamic fields for the start and end time, the calendar shows the event.

Click on an entry to see the details of the ticket. An event information box will be displayed with the event items, that an agent has set for the ticket.

Click on the *Open Ticket* button to open the ticket.

See also:

Home ▶ Ticket Calendar

Calendar

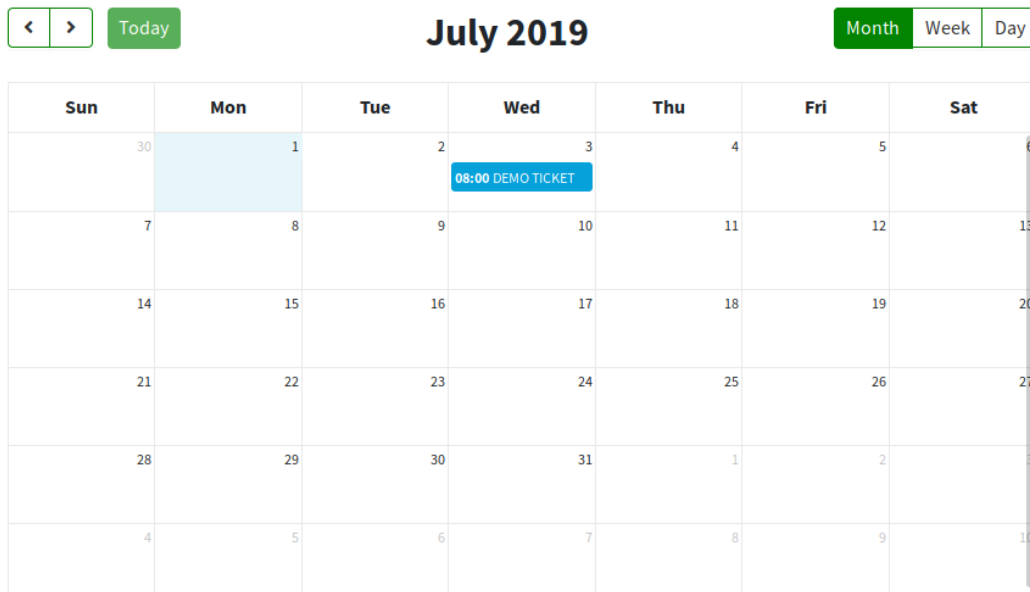


Fig. 1: Events Ticket Calendar Widget

This screen is not added to any menu of the external interface by default. To create a link to the *Ticket Calendar* screen, an administrator needs to add `/ticket/calendar` 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

The image shows a 'Ticket Information' popover with a blue header and a close button (X) in the top right. Below the header are several rows of information, each with a light gray background:

- Customer ID:** stark-industries
- Customer user:** ironman
- Event Description:** Demo Ticket - Raw
- Priority:** 3 normal
- Queue:** Raw
- State:** open
- Ticket Number:** 2019070110000011
- Title:** Demo Ticket
- Type:** Unclassified

Below these fields is a section titled 'Dynamic Fields' with two rows:

- Calendar End Time:** 07/03/2019 07:00 (UTC)
- Calendar Start Time:** 07/03/2019 06:00 (UTC)

At the bottom of the popover is a green button labeled 'Open Ticket'.

Fig. 2: Ticket Details Popover

Customer Frontend Link Object

With this feature add-on, linked FAQs, 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 FAQ 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 FAQ 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

21.1 Administrator Interface

This package has no administrator interface.

21.2 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 FAQ article. Make sure that the FAQ article is visible to customers or public. Just FAQ articles of this type can be displayed in the external interface.
5. Link the FAQ article to the ticket created for the customer user.

21.3 External Interface

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

To use this functionality, an agent has to create tickets and FAQ articles in *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 FAQ article.
4. See the widget containing the linked objects in the right sidebar of the ticket detail view.

Clicking on a ticket or FAQ article will lead you to the linked ticket or FAQ 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

22.1 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.

22.1.1 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.

The screenshot shows the 'Allocate Services to Customer' screen. The breadcrumb navigation is 'Manage Customer-Service Relations' > 'Allocate Services to Customer '<DEFAULT>'. The main content area is a table with the following data:

SERVICE	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"/>

At the bottom of the table, there are buttons for 'Save', 'Save and finish', and 'Cancel'. The left sidebar contains an 'Actions' section with a 'Go to overview' button, a 'Filter for Services' section with a search input, and a 'Hint' section with the text: 'A customer can be associated with many services and also a service can be related with more than one customers.'

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.

22.2 Agent Interface

After installation of the package a new widget will be available in the *Customer Information Center* screen of the agent interface.

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

22.2.1 Customers

After the installation of the package, a new widget *Services* appears in the *Customer Information Center* screen in the agent interface. In this widget all assigned services and the amount of open and closed tickets are displayed.

Customer Information Center

After installation of the package, a new content widget *Services* will be available.

Customer Information Center Content Widgets

The *Services* widget displays all assigned services and the amount of open and closed tickets.

Services		
SERVICE	OPEN	CLOSED
Backup and Archiving	0	0
File / Print	0	0
Internet	1	0

Fig. 5: Services Widget

22.3 External Interface

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

22.3.1 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 *Internal News* 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 box in the agent dashboard. The dashboard news 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

23.1 Administrator Interface

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

23.1.1 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.

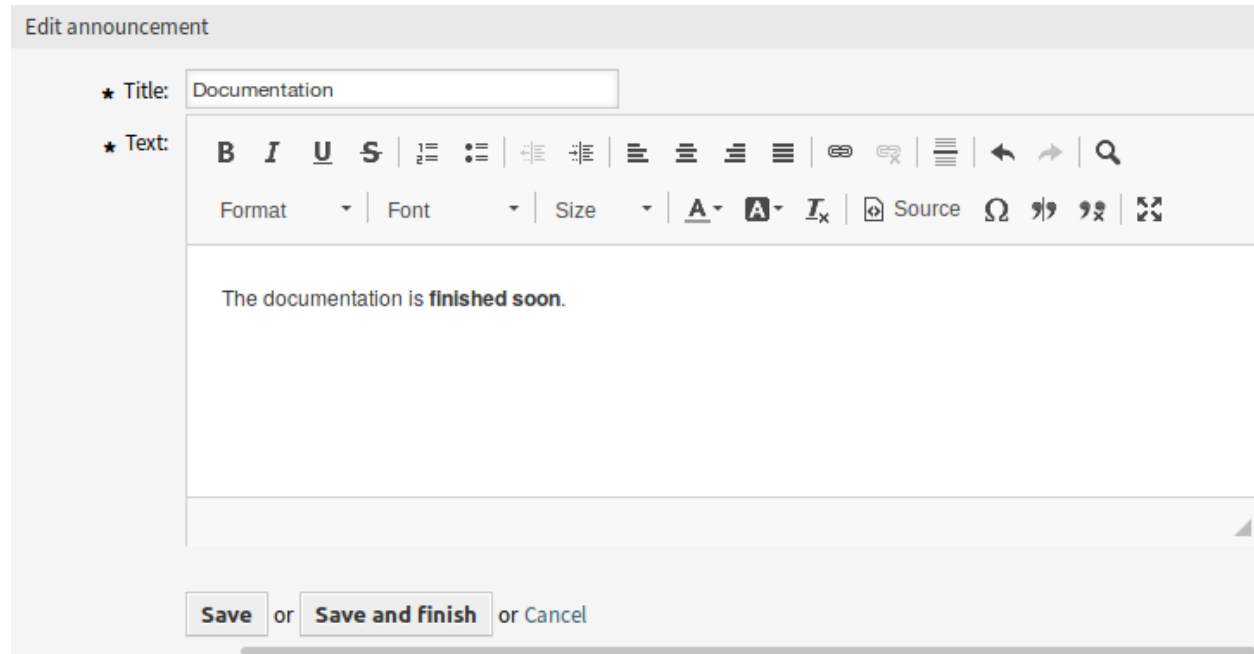


Fig. 5: Edit Announcement Screen with Rich Text Editor

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.

23.2 Agent Interface

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

23.2.1 Personal Menu

There is an optional setting to show a preference widget in the agent personal preferences.

Personal Preferences

Use this screen to set announcement overview limit in the agent personal preferences. The personal preferences screen is available in the *Personal Preferences* menu item of the *Avatar* menu.

See also:

The `PreferencesGroups###DashboardNewsOverviewPageShown` setting needs to be activated to use this feature.

Miscellaneous

These settings are for change some visual appearance of **OTRS**.

Announcement Overview Limit

In this widget can be selected the number of displayed announcements per page.

Fig. 6: Announcement Overview Limit Widget

Select a number for displayed announcements from the drop-down list.

23.2.2 Dashboard

After installation of the package a new widget will be displayed in the agent dashboard. This widget also has an overview screen and a details screen for announcements.

Dashboard

After installation of the package a new widget will be displayed in the right sidebar of the agent dashboard.



Fig. 7: Internal News Widget

Clicking on an entry in the widget will open the *Announcement details* screen.

There is a button to go back to the overview screen to see all announcements.

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.

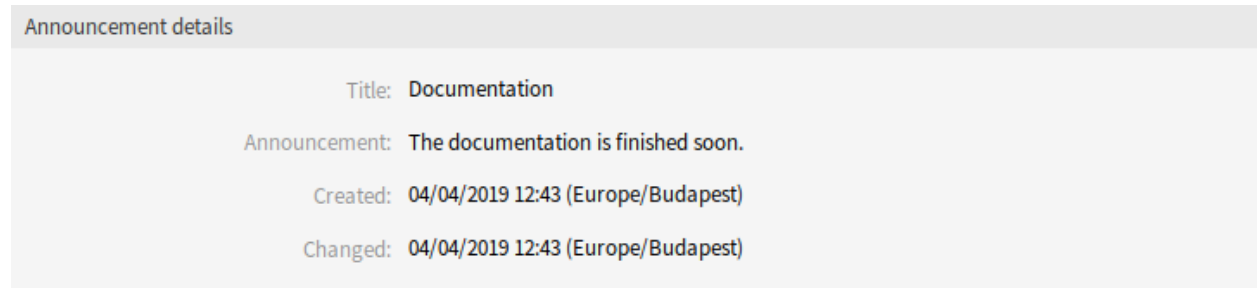


Fig. 8: Announcement Details Screen

TITLE	ANNOUNCEMENT	CHANGED	CREATED
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. 9: Announcement Overview Screen

23.2.3 Dashboard News

After installation of the package a new menu will be available in the agent interface.

Overview

This menu item opens the *Dashboard News Management* screen of the *Dashboard News* administration module.

Add Announcement

This menu item opens the *Add Announcement* screen of the *Dashboard News* administration module.

23.3 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

24.1 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.

24.1.1 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*.

24.1.2 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
```

24.1.3 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
- CreatedTypeIDs
- 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* (\geq) 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

24.2 Agent Interface

This package has no agent interface.

24.3 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

25.1 Administrator Interface

This package has no administrator interface.

25.1.1 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.

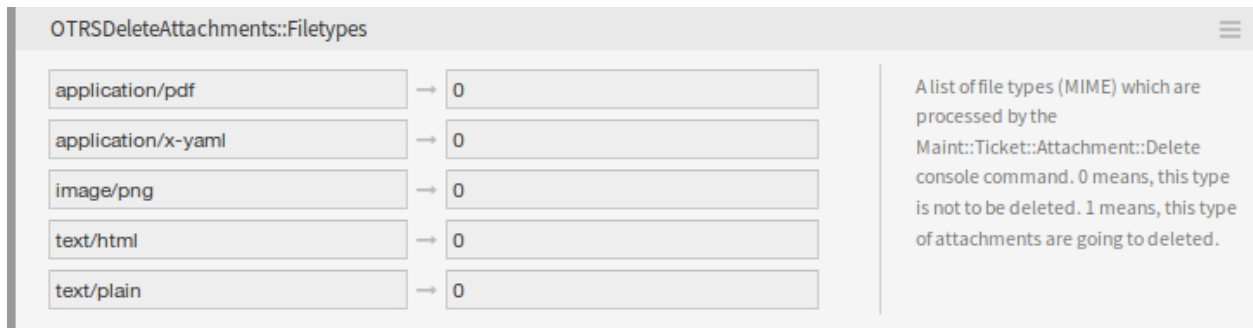


Fig. 1: File Types Setting

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:

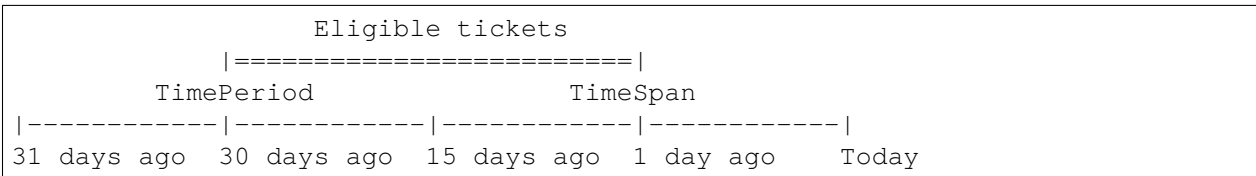
OTRSDeleteAttachments::Queues A list of queues (specified by their names) which are processed by the `Maint::Ticket::Attachment::Delete` console command.

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.

25.2 Agent Interface

This package has no agent interface.

25.3 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

26.1 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.

26.1.1 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 [administrator manual](#).

Dynamic Field Settings

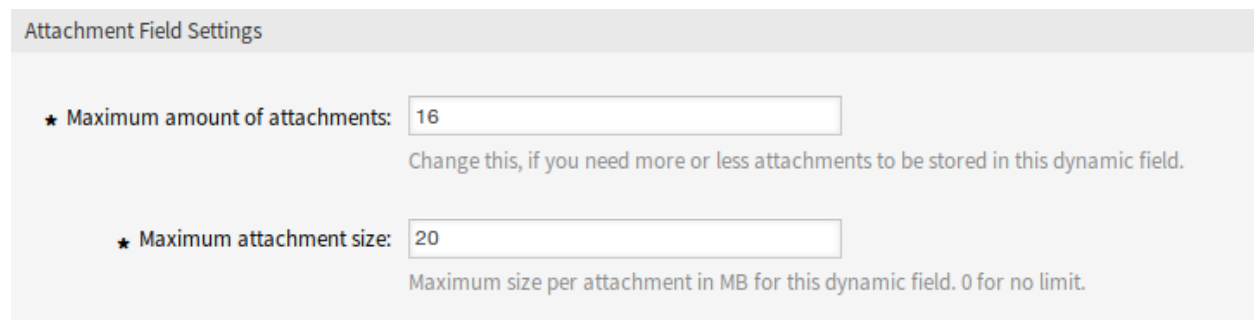
The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.

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.



The image shows a screenshot of the 'Attachment Field Settings' form. It has a light gray header with the title 'Attachment Field Settings'. Below the header, there are two settings, each with a mandatory asterisk icon and a text input field. The first setting is 'Maximum amount of attachments:' with a value of '16'. Below it is a blue italicized note: 'Change this, if you need more or less attachments to be stored in this dynamic field.' The second setting is 'Maximum attachment size:' with a value of '20'. Below it is a blue italicized note: '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.

26.2 Agent Interface

This package has no agent interface, but the dynamic field of type attachment can be added to many screens.

26.2.1 Upload Files

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 *Submit* 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.

Uploading multiple files with identical names to one dynamic field of type attachment will result in an error notifying the agent that this is not permitted.

26.2.2 Download Files

The dynamic field of type attachment is displayed in *Ticket Zoom* screen, if it is added to the screen and a file has been uploaded.

To download a file from a ticket dynamic field of type attachment:

1. Find the *Attachments* link in the *Ticket Information* widget in the right sidebar.
2. Click on the *Attachments* link. A small *Attachments* dialog appears showing all filenames as well as their size.
3. Click on a filename to download the file.

To download a file from an article dynamic field of type attachment:

1. Select an article, that has a dynamic field of type attachment with at least one uploaded file.
2. Expand the article header with the *i* icon in the top right corner of the article widget.
3. Find the *Attachments* link in the article header.
4. Click on the *Attachments* link. A small *Attachments* dialog appears showing all filenames as well as their size.
5. Click on a filename to download the file.

26.2.3 Delete Files

If a ticket dynamic field of type attachment has been configured to be displayed in the *Ticket Free Fields* screen, 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 *Ticket Free Fields* screen.
2. Click on the *Miscellaneous* → *Free Fields* menu item in the ticket menu.
3. Click on the trash 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.

26.2.4 Search for Files

Dynamic field of type attachment supports searching for the file names.

To search for a dynamic field of type attachment:

1. Make sure that the dynamic field of type attachment is added to the *Ticket Search* screen.
2. Click on the *Tickets* → *Search* menu item in the main menu.
3. Add the dynamic field to the search criteria. The * character as wildcard is supported.

26.3 External Interface

This package has no external interface.

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

27.1 Administrator Interface

This package contains a configuration interface to create dynamic fields of the type *Calculation*. These fields can perform simple mathematical operations.

27.1.1 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:

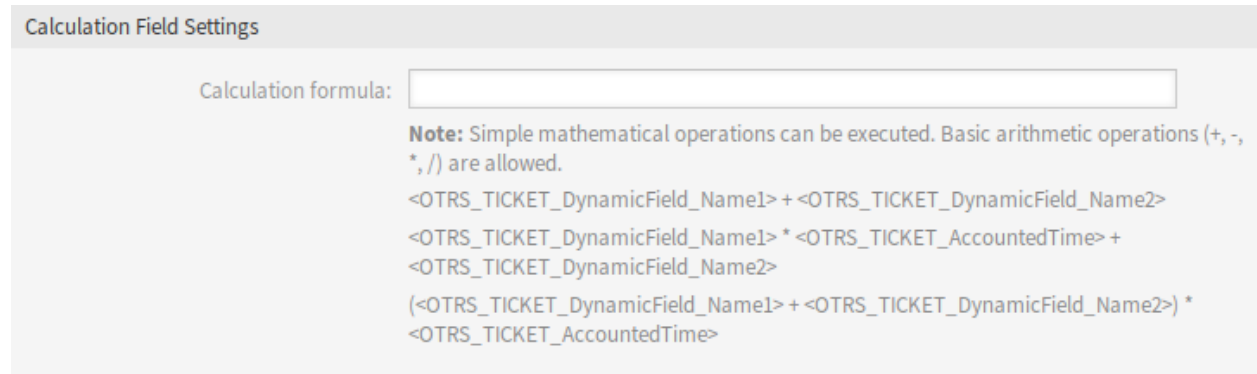
The usage of dynamic fields and the general dynamic field settings are described in the [administrator manual](#).

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.

27.2 Agent Interface

This package has no agent interface, but the dynamic field of type calculation can be added to many screens.

27.3 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/products (CIs)

- 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 *ITSM Configuration Management* feature.

Note: Not compatible with the following feature add-ons:

- *Ticket Forms*
-

28.1 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.

28.1.1 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 [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.

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 by CustomerID Select if the available configuration items should be restricted by customer in agent and/or external interface.

CustomerID attribute The name of the configuration item field containing the customer. Usually this is the owner field.

Value lookup for CustomerID attribute Select if the readable value for the customer ID field should be used for restriction. Select this if the values of attribute are defined by a general catalog class.

Restriction by CustomerUserID Select if the available configuration items should be restricted by customer user in agent and/or external interface.

CustomerUserID attribute The name of the configuration item field containing the customer user. Usually this is the owner field.

Value lookup for CustomerUserID attribute Select if the readable value for the customer user ID field should be used for restriction. Select this if the values of attribute are defined by a general catalog class.

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.

Configuration Item Field Settings

Filter Options

Class:
 Filter for config item classes.

Deployment State:
 Filter for config item deployment states.

Incident State:
 Filter for config item incident states.

Restriction by CustomerID:
 Select if the available config items should be restricted by CustomerID in agent and/or external interface.

CustomerID attribute:
 The name of the config item field containing the CustomerID.

Value lookup for CustomerID attribute:
 Select if the readable value for the CustomerID field should be used for restriction.

Restriction by CustomerUserID:
 Select if the available config items should be restricted by CustomerUserID in agent and/or external interface.

CustomerUserID attribute:
 The name of the config item field containing the CustomerUserID.

Value lookup for CustomerUserID attribute:
 Select if the readable value for the CustomerUserID field should be used for restriction.

Link Options

Link type:
 Type of automatically created links for config items. 'None' disables linking.

Link direction:
 Please select a possible direction for selected link type.

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:

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 CustomerID* or *Restriction by CustomerUserID* is deactivated.

Follow dynamic field attribute The name of the configuration item field containing the dynamic field value.

Disable default values If activated, the dynamic field will not display the default values.

See also:

The actions where the option will be active can be configured in the following system configuration setting:

- `DynamicFieldConfigItem::DisableDefaultValues`

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

28.2 Agent Interface

This package has no agent interface, but the dynamic field of type configuration item can be added to many screens.

28.2.1 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 Zoom* screen. 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.

28.3 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

29.1 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.

29.1.1 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 [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:

Searchfield

Listfield

Add value:

Show link:

Here you can specify an optional HTTP link for the field value in Overviews and Zoom screens.
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 zoom. 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:

CacheTTL:

Searchprefix:

Searchsuffix:

Result Limit:

29.1. Administrator Interface:

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 in ticket overviews and zoom screens. 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 zoom. 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.

CacheTTL This value defines the period of validity of the database cache in seconds. Equal queries to the database will be answered through the cache (local file system) within this period instead of asking the database again.

Searchprefix This value will be put in the front of every search term while using the 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.

Searchsuffix 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.

29.2 Agent Interface

This package has no agent interface, but the dynamic field of type database can be added to many screens.

29.2.1 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.

29.3 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 Search Booster

For the ticket search in the agent or external interface, existing values of the database are collected in dynamic fields. In highly frequented systems with a large amount of tickets and/or dynamic fields, the high number of entries that have to be searched can cause a considerable delay in the search process. Search results in these systems can only be cached for a short time and then the database inquiry has to be done again.

This feature add-on keeps track of the values used for all relevant dynamic fields and their frequency. In the event of a new inquiry, the feature add-on can now provide the data very quickly. Unnecessary load times are reduced considerably.

Benefits

- Fast database inquiries.
- System performance improvements.

Target Groups

- Call centers
- Customer service
- Sales
- Order management
- Complaint management

Available in Service Package

- PLATINUM

Package Name in OTRS Package Manager

- OTRSDynamicFieldSearchBooster

30.1 Administrator Interface

This package has no administrator interface. It provides performance optimizations for the ticket search for some heavily used systems.

When using the ticket search in agent or external interface, historical values will be gathered from the database for select-type dynamic fields (e.g. drop-down and multiselect). On systems with a large number of tickets and/or dynamic fields, looking through potentially millions of entries can take more time than desirable which leads to a noticeable delay until the form has opened.

The result is cached but as dynamic fields change constantly on heavily used systems, caches are outdated quite frequently and the database has to be queried again. This package keeps track of used values and their number of occurrence for all relevant dynamic fields and uses this data to present a quick response when historical values are requested.

On package installation the current state for all dynamic fields will be retrieved and afterwards it is automatically being updated. However, synchronization can be force triggered by a new console command `Maint::Ticket::DynamicFieldSearchBoosterSync`.

The search booster is initialized on package installation (via a daemon task). It might take a while for this task to complete. Until that time historical values retrieved in the search form may be incorrect.

The search booster is in effect automatically when the ticket search form is opened and should deliver the same results as before.

30.2 Agent Interface

This package has no agent interface.

30.3 External Interface

This package has no external interface.

Dynamic Field Value Import

This feature add-on offers additional options for uploading values from CSV files for dynamic fields. The data from individual columns of the CSV file are assigned to a specific dynamic field that you can flexibly select. These files can additionally contain headers into which you can insert the desired name of the target fields. Before the value can be imported, the target fields have to be created manually.

After a successful upload, the *Change Overview* widget shows all new or deleted values for the dynamic fields affected. These values are marked in colors.

In addition, the feature add-on enables the automatic creation of ACLs based on the dynamic fields chosen and their imported values. This significantly reduces the manual workload for ACL creation.

Dependencies between the dynamic fields are created from left to right. If values for more than two dynamic fields are imported, the allocation of the ACL relations starts in the column furthest to the left. Multi-level dependencies are therefore possible.

Benefits

- Fast import of modified product portfolios.
- Customer service and sales are more quickly up-to-date.
- Easier integration of partners from outsourced corporate areas.

Target Groups

- Call centers
- Customer service
- Wholesalers
- Sales
- Order management

Available in Service Package

- GOLD

Package Name in OTRS Package Manager

- OTRSDynamicFieldValueImport

31.1 Administrator Interface

This package provides functionality to import dynamic field possible values from CSV files. Furthermore it is possible to create left → right associated access control lists (ACLs) between these dynamic fields and their imported values.

31.1.1 Processes & Automation

This package makes it possible to import values for dynamic fields and adds a new widget into the left sidebar of the *Dynamic Fields Management* screen.

Dynamic Fields

After installation of the package a new widget *Value Import* will be available in the left sidebar of the *Dynamic Fields Management* screen.

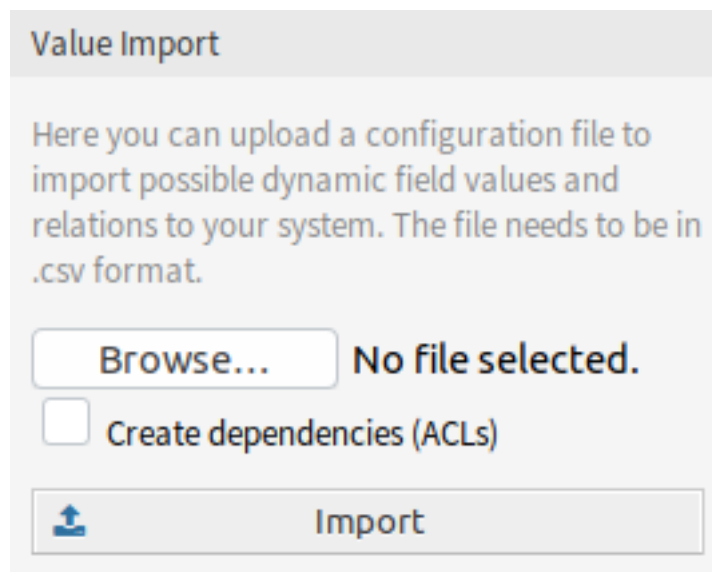


Fig. 1: Value Import Widget

To import values for a dynamic field:

1. Make sure that the target dynamic fields are already created.
2. Click on the *Browse...* button of the *Value Import* widget in the left sidebar.
3. Select a proper formatted `.csv` file.
4. Select the *Create dependencies (ACLs)* checkbox.
5. Click on the *Import* button.
6. Select the target dynamic fields and see the change difference.
7. Modify the ACL settings, if needed.

8. Click on the *Submit* button.

Example Usage

Target dynamic fields have to be created manually before they can be used for value import. Only select-type fields are allowed (drop-down and multiselect).

Create the following dynamic fields:

Object	Type	Name	Label	Possible values
Ticket	Multiselect	Manufacturer	Manufacturer	Leave it empty
Ticket	Multiselect	Car	Car	Leave it empty
Ticket	Multiselect	ModelYear	Model Year	Leave it empty

To guarantee a proper working of the feature provided by this package, please format your CSV files according to the example. The first line containing dynamic field names is optional, but the dynamic fields will be auto-selected in the next screen, if they are provided.

By default the separator defined in your language will be used (usually , or ;). Alternatively you can override this by using the CSV separator user preference.

```
Manufacturer, Car, ModelYear
BMW, X3, 2016
BMW, 318d, 2016
BMW, 318d, 2017
BMW, 320d, 2016
Audi, A4, 2015
Audi, A4, 2016
Audi, A5, 2015
Audi, A5, 2016
Audi, A6, 2016
```

The CSV file contains information about possible values for dynamic fields. The first header line contains the target dynamic field names.

Browse for the CSV file and check the *Create dependencies (ACLs)* box if you want to generate ACLs. Click on the *Import* button, a new screen will be opened.

In this overview you will see a table indicating CSV column, the selected target dynamic field, a colored change difference between old and new possible values and (if ACL setting is enabled) a preview of the relations between the fields. This helps comparing the data before the values are written.

Data from every CSV column will be assigned to a specific dynamic field. The target field can be chosen. If a header line is used, the field names are pre-selected.

Warning: The existing possible values for all affected fields will be completely overwritten with the uploaded values. Therefore you should carefully check if the indication of removed and new values for each dynamic field matches your expectation. In order to retain existing values they need to be contained in the uploaded CSV file.

It is possible to automatically create left → right associated ACLs based on the selected dynamic fields and their imported values. This will reduce the effort to creating ACLs manually. The dependencies between dynamic fields and values are created from the left to right side.

▼ Change Overview

COLUMN #	TARGET DYNAMIC FIELD	DIFF OF POSSIBLE VALUES	ACL RELATIONS BETWEEN FIELDS															
1	Manufacturer x	+Audi +BMW																
2	Car x	+318d +320d +A4 +A5 +A6 +X3	<table border="1"> <thead> <tr> <th>MANUFACTURER</th> <th>CAR</th> </tr> </thead> <tbody> <tr> <td>▪ Audi</td> <td>▪ A4 ▪ A5 ▪ A6</td> </tr> <tr> <td>▪ BMW</td> <td>▪ 318d ▪ 320d ▪ X3</td> </tr> </tbody> </table>	MANUFACTURER	CAR	▪ Audi	▪ A4 ▪ A5 ▪ A6	▪ BMW	▪ 318d ▪ 320d ▪ X3									
MANUFACTURER	CAR																	
▪ Audi	▪ A4 ▪ A5 ▪ A6																	
▪ BMW	▪ 318d ▪ 320d ▪ X3																	
3	ModelYear x	+2015 +2016 +2017	<table border="1"> <thead> <tr> <th>MANUFACTURER</th> <th>CAR</th> <th>MODELYEAR</th> </tr> </thead> <tbody> <tr> <td>▪ Audi</td> <td>▪ A4 ▪ A5</td> <td>▪ 2015 ▪ 2016</td> </tr> <tr> <td>▪ Audi</td> <td>▪ A6</td> <td>▪ 2016</td> </tr> <tr> <td>▪ BMW</td> <td>▪ 318d</td> <td>▪ 2016 ▪ 2017</td> </tr> <tr> <td>▪ BMW</td> <td>▪ 320d ▪ X3</td> <td>▪ 2016</td> </tr> </tbody> </table>	MANUFACTURER	CAR	MODELYEAR	▪ Audi	▪ A4 ▪ A5	▪ 2015 ▪ 2016	▪ Audi	▪ A6	▪ 2016	▪ BMW	▪ 318d	▪ 2016 ▪ 2017	▪ BMW	▪ 320d ▪ X3	▪ 2016
MANUFACTURER	CAR	MODELYEAR																
▪ Audi	▪ A4 ▪ A5	▪ 2015 ▪ 2016																
▪ Audi	▪ A6	▪ 2016																
▪ BMW	▪ 318d	▪ 2016 ▪ 2017																
▪ BMW	▪ 320d ▪ X3	▪ 2016																

Fig. 2: Change Overview Widget

▼ ACL settings

Create ACLs?

Left → right depending ACLs between affected fields will be created automatically on import.

ACL Prefix:

Given prefix is used for all created ACLs. Change prefix to affect execution order (interoperability with manually created ACLs).

Remove ACLs with other prefixes:

All previously auto-generated ACLs will be removed.
Note: If a different prefix was used previously and this option is not used, old ACLs will remain and might cause undesired behavior.

Auto deploy:

Automatically deploy ACLs after creation.
Note: This will also affect other ACLs that have not been deployed yet.

Fig. 3: ACL Settings Widget

The following settings are available when using this functionality.

Create ACLs? If you want to generate ACLs, check this box. Left → right depending ACLs between affected fields will be created automatically on import.

ACL Prefix Given prefix is used for all created ACLs. Change prefix to affect execution order (interoperability with manually created ACLs).

Remove ACLs with other prefixes Select this checkbox, if all previously auto-generated ACLs should be removed.

Note: If a different prefix was used previously and this option is not used, old ACLs will remain and might cause undesired behavior.

Auto deploy Select this checkbox, if you would like to automatically deploy ACLs after creation.

Note: This will also affect other ACLs that have not been deployed yet.

The ACL creation is implemented in the following way:

- If the dynamic field with the value(s) from the left side is matched, it is (only) possible to select the dynamic field with the value(s) from the right side.
- If values for more than two dynamic fields are imported (3 or more columns exist in the CSV file), ACL relations are always created starting from the leftmost column. This allows for multi-level dependencies.

After you have finished all checks, just submit the form. The values of the affected dynamic fields will be updated in the background and, if activated, the ACLs will be generated. Afterwards you will be redirected to the dynamic field overview again.

Now go to the *ACL Management* screen, and check the generated ACLs.

















ACL NAME	COMMENT	VALIDITY	EXPORT	COPY
9999-DynamicFieldValueImport-Manufacturer-Car-000	Created automatically by OTRSDynamicFieldValueImport.	valid		
9999-DynamicFieldValueImport-Manufacturer-Car-001	Created automatically by OTRSDynamicFieldValueImport.	valid		
9999-DynamicFieldValueImport-Manufacturer-Car-002	Created automatically by OTRSDynamicFieldValueImport.	valid		
9999-DynamicFieldValueImport-Manufacturer-Car-ModelYear-000	Created automatically by OTRSDynamicFieldValueImport.	valid		
9999-DynamicFieldValueImport-Manufacturer-Car-ModelYear-001	Created automatically by OTRSDynamicFieldValueImport.	valid		
9999-DynamicFieldValueImport-Manufacturer-Car-ModelYear-002	Created automatically by OTRSDynamicFieldValueImport.	valid		
9999-DynamicFieldValueImport-Manufacturer-Car-ModelYear-003	Created automatically by OTRSDynamicFieldValueImport.	valid		
9999-DynamicFieldValueImport-Manufacturer-Car-ModelYear-004	Created automatically by OTRSDynamicFieldValueImport.	valid		

Fig. 4: Generated ACLs for Dynamic Field Restrictions

Limitations

However this package makes it easy to import values for dynamic fields and generate the appropriate ACLs, there are some limitations.

- Dynamic field creation and configuration has to be done manually (e.g. to define if a field should be a drop-down or multiselect and if empty values are allowed).
- ACLs are created in a way that secondary fields require a selection on their primary field before any selection is possible.
- ACLs will always be created for all dynamic fields from the CSV file. Unrelated dynamic fields have to be imported separately.

31.2 Agent Interface

This package has no agent interface.

31.3 External Interface

This package has no external 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

32.1 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 *Ticket Free Fields* screen), the details of the already created ticket are also included in the request.

32.1.1 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 [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.

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.

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 in overviews and zoom screens. Optional HTTP link works only for single-select fields.
Example: `http://some.example.com/handle?query=[% Data.Field1 | uri %]`

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

Fig. 1: Web Service Dynamic Field Settings

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 in overviews and zoom screens. Optional HTTP link works only for single-select fields. Example:

```
https://some.example.com/handle?query=[% Data.Field1 | uri %]
```

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 zoom. 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 *Ticket 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 unmached 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">

  <xsl:output method="xml" encoding="utf-8" indent="yes" />
```

(continues on next page)

(continued from previous page)

```

<!-- 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: ''
      IncludeTicketData: ''
      MappingInbound: {}
      MappingOutbound: {}

```

(continues on next page)

(continued from previous page)

```

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
↪txt.
          did not receive this file, see https://www.gnu.org/licenses/gpl.

          -->

          <!-- DOCUMENTATION

          * Example XML Input *
          <RootElement>
            ...
          </RootElement>

          * Example XML Output *
          <RootElement>
            <PossibleValues>
              <Key>???

```

(continues on next page)

(continued from previous page)

```

-->

<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
the enclosed file COPYING for license information (GPL). If you
did not receive this file, see https://www.gnu.org/licenses/gpl.
→txt.

```

(continues on next page)

(continued from previous page)

```

-->

<!-- DOCUMENTATION

* Example XML Input *
<RootElement>
    ...
</RootElement>

* Example XML Output *
<RootElement>
    <PossibleValues>
        <Key>???

```

Type: XSLT**Type:** Generic::PassThrough**Transport:****Config:**

(continues on next page)

(continued from previous page)

```
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'
```

32.2 Agent Interface

This package has no agent interface.

32.3 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

33.1 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.

33.1.1 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.

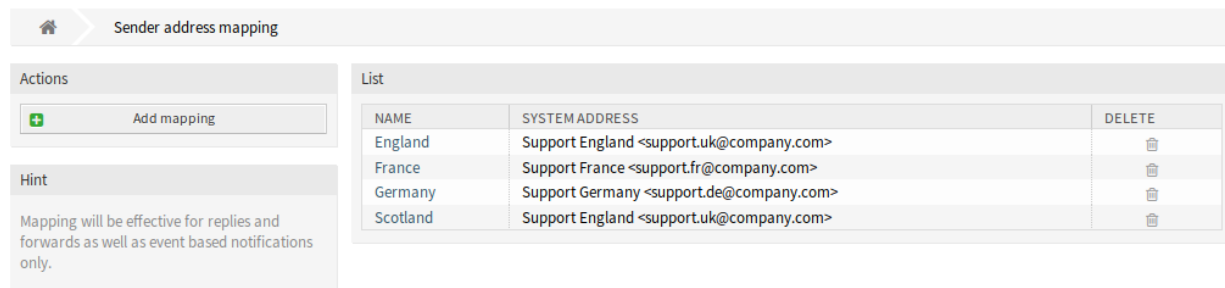


Fig. 1: Sender Address Mapping Management Screen

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.

33.1.2 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.

33.2 Agent Interface

This package has no agent interface.

33.2.1 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

33.2.2 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).

33.3 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

Note: Not compatible with the following feature add-ons:

Advanced Escalations

34.1 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).

34.1.1 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.

34.1.2 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
```

34.2 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 Zoom* screen for the newly created ticket. Notice that the escalation is running.

3. Once the escalation is running, change the ticket state to one that has been configured in `EscalationSuspendStates` setting.
4. Go back to the *Ticket Zoom* screen for the suspended escalation ticket and verify that the escalation is not running.
5. Change the ticket state to one different from those configured on `EscalationSuspendStates` setting. The escalation continues.

On ticket menu will be located a new option for *Escalation view - Without Suspend State*. Open this view and check the ticket without a suspend state.

34.3 External Interface

This package has no external interface.

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

35.1 Administrator Interface

This package has no administrator interface.

35.2 Agent Interface

This module extends the statistic feature with adding new modules for ticket analysis. The package also adds a new ready to use statistic.

35.2.1 Reports

This package contains additional statistic modules for ticket analysis.

Statistics

After installation of the package, a new statistic and two new dynamic list objects will be available in the *Statistics* screen.

New Statistic

After installation of the package, a new statistic will be available in the *Statistics* 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 *Statistics* widget.
2. Click on the *Run now* icon in the last column.
3. Select the time frame and the queues to include in the statistic.
4. Generate your statistic.

New Dynamic Objects

After installation of the package, two new dynamic objects will be added to the system.

Statistics

1-12 of 12

▲ STAT#	TITLE	OBJECT	EXPORT	DELETE	RUN
10001	List of tickets closed last month	Ticketlist			
10002	New Tickets	TicketAccumulation			
10003	List of open tickets, sorted by time left until response deadline expires	Ticketlist			
10004	List of tickets closed, sorted by response time.	Ticketlist			
10005	List of tickets created last month	Ticketlist			
10006	List of the most time-consuming tickets	Ticketlist			
10007	List of open tickets, sorted by time left until escalation deadline expires	Ticketlist			
10008	List of tickets closed, sorted by solution time	Ticketlist			
10009	Overview about all tickets in the system	TicketAccumulation			
10010	List of open tickets, sorted by time left until solution deadline expires	Ticketlist			
10011	Changes of status in a monthly overview	StateAction			
10012	Response Statistic	OTRSStatsResponseStatistic			

Fig. 1: Statistics Overview Screen

Settings

Object: OTRSStatsResponseStatistic

Description: A statistic to count the responses used, i.e. how many times has the response "Delivery Missing" be used in the last 30 days.

Format:

Configurable Params of Static Stat

Start Day:

Start Month:

Start Year:

End Day:

End Month:

End Year:

Queues:

or Cancel

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 dynamic object:

1. Create a new *Dynamic List* statistic.
2. Choose the dynamic 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 dynamic object:

1. Create a new *Dynamic List* statistic.
2. Choose the dynamic object `OTRSStatsQueueWorkingTime`.
3. Click on the *X-axis* button and 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.

35.3 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

Note: Not compatible with the following feature add-ons:

- *Advanced Editor*
 - *Categories for Text Modules*
 - *CLs in Customer Frontend*
 - *Restore Pending Information*
 - *Ticket Forms*
-

36.1 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 Phone Ticket
- New Email Ticket
- Ticket Phone Inbound
- Ticket Phone Outbound
- Ticket Note
- Ticket Close
- Ticket Move
- Ticket Pending
- Ticket Free Fields
- Ticket Owner
- Ticket Responsible
- Ticket Priority

External interface:

- New Ticket
- Ticket Reply (within *Ticket Detail View*)

In the *Ticket Search* screen the functionality is limited to show or hide dynamic fields if non-ticket-specific ACLs are used (e.g. *Properties* → *User* → *UserID* or *Roles*).

36.1.1 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
226			Chapter 36. Hide/Show Dynamic Fields	
Ticket	Textarea	Remarks	Remarks	
Ticket	Date	RegistrationDate	Registration Date	
Ticket	Date	InvoiceDate	Invoice Date	

Add the dynamic fields to the *New Phone Ticket* screen via setting `Ticket::Frontend::AgentTicketPhone###DynamicField:`

```
Brand → 1 - Enabled
VWModel → 1 - Enabled
VWProductionFacility → 1 - Enabled
PeugeotModel → 1 - Enabled
PeugeotProductionFacility → 1 - Enabled
Fuel → 1 - Enabled
Accessories → 1 - Enabled
Remarks → 1 - Enabled
RegistrationDate → 1 - Enabled
InvoiceDate → 1 - Enabled
```

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:
        - CD Radio
        - Climate Control
      DynamicField_Fuel:
        - Gasoline
      DynamicField_VWProductionFacility:
        - Bratislava
    PossibleAdd:
      Form:
        - Accessories
        - Fuel
        - Remarks
        - RegistrationDate
        - InvoiceDate
    PossibleNot:
      Form:
        - PeugeotModel
        - PeugeotProductionFacility
  ConfigMatch:
    Properties:
      Ticket:
        DynamicField_Brand:
          - VW
        DynamicField_VWModel:
          - Up
CreateBy: root@localhost
CreateTime: 2019-07-22 11:47:02
Description: ''
ID: 2
Name: ACL-VW-Up

```

(continues on next page)

(continued from previous page)

```

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:
- CD Radio
- Climate Control
DynamicField_Fuel:
- Gasoline
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.

36.2 Agent Interface

This package has no agent interface.

36.3 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

37.1 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.

37.1.1 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.

37.1.2 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.

37.1.3 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>
```

37.2 Agent Interface

This package has no agent interface.

37.3 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

38.1 Administrator Interface

This package has no administrator interface.

38.1.1 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.

38.2 Agent Interface

This package has no agent interface.

38.3 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

39.1 Administrator Interface

This package has no administrator interface, but the feature add-on adds a new communication channel that is selectable for articles.

39.1.1 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): [Text input field]
- Description (long): [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 *New Process Ticket* and start a new process.

39.2 Agent Interface

This package has no agent interface, but the feature add-on adds the regular email fields to *New Process Ticket* screen.

39.2.1 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 *New Process Ticket* menu item of the *Tickets* 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.

39.3 External Interface

This package has no external interface.

Queue Responsible

In service management, results depend on how tasks are assigned and making sure that no service request stays unanswered. Therefore, **OTRS** allows you to assign a responsible person to a ticket who takes care of solving it. However, in companies with many regional branches or many levels of hierarchy in their service management staff, there is a complex queue structure. This can make the responsibilities unclear and lead to meaningless reporting.

This feature add-on is the solution, because it allows you to assign a responsible party at the queue level.

Benefits

- Clear separation of divisions by queue responsables.
- More significant reporting.

Target Groups

- IT service management
- Call centers
- Sales
- Logistics

Available in Service Package

- GOLD

Package Name in OTRS Package Manager

- OTRSQueueResponsible

40.1 Administrator Interface

This package has no administrator interface.

40.1.1 Configuration

This package provides the responsible feature for queues and handles the `TicketQueueUpdate` and `TicketCreate` events to set the responsible based on the name of the queue.

To activate the queue responsible feature:

1. Go to *System Configuration* screen.
2. Make sure that `Ticket::Responsible` setting is enabled in order to be able to check it on *Ticket Zoom* screen.
3. Select *OTRSQueueResponsible* in the *Navigation* widget.
4. Navigate to *Core* → *Ticket* → *OTRSQueueResponsible* in the navigation tree.
5. Set values for `Ticket::QueueResponsible` setting. Key is the queue name as stored in database (e.g. *Junk*, *Misc* or *Raw*) and content is the login name of the user to be defined as responsible.
6. Modify `Ticket::QueueResponsibleOnNewTicket` setting if needed. If the value is set to *Yes* then also the responsible of new tickets is set per queue and the new tickets are processed like the change of queue event.

40.2 Agent Interface

This package has no agent interface.

40.3 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 *ITSM Configuration Management* feature.

41.1 Administrator Interface

This package has no administrator interface, but it adds the possibility to import Ready2Adopt processes.

41.1.1 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.

41.2 Agent Interface

This package has no agent interface.

41.3 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

42.1 Administrator Interface

This package has no administrator interface, but it adds the possibility to import Ready2Adopt processes.

42.1.1 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.

42.2 Agent Interface

This package has no agent interface.

42.3 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.

43.1 Administrator Interface

This package has no administrator interface, but it adds the possibility to import Ready2Adopt web services.

43.1.1 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.

43.2 Agent Interface

This package has no agent interface.

43.3 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

Note: Not compatible with the following feature add-ons:

- *Hide/Show Dynamic Fields*

- *Ticket Forms*
-

44.1 Administrator Interface

This package has no administrator interface.

44.2 Agent Interface

This package has no agent interface, the feature add-on uses the existing ticket screens.

44.2.1 Tickets

On opening the *Ticket Pending* screen 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 ticket pending pop-up screen:

- Pending date
- State
- Title
- Text (if the text comes from a note article created by the *Ticket Pending* screen)

The functionality works only, when the target time is not expired, then the standard functionality sets the usual default values.

Ticket Zoom

After installation of the package, the functionality is automatically integrated in the *Ticket Pending* screen. If there was an unexpired pending time set before, the *Ticket Pending* screen is pre-filled with the information of the last action where the pending time was set.

Use Case Example

1. The agent sets a *pending* state with a pending time and other information like subject and text in the ticket.
2. The pending time is stored now in the *Ticket Information* widget in the right sidebar of the *Ticket Zoom* screen.
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 *Ticket Pending* screen easily.
5. When the agent opens the *Ticket Pending* screen, 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 *Ticket Pending* screen.

The text field is only pre-filled from note articles, since this is the type of article the *Ticket Pending* screen creates.

44.3 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

Note: Not compatible with the following feature add-on:

- *VIP Customer*
-

45.1 Administrator Interface

This package offers the possibility to restrict the access of certain agents to data of certain customers following a mapping table.

45.1.1 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.

Manage PartnerID-CustomerID Relations

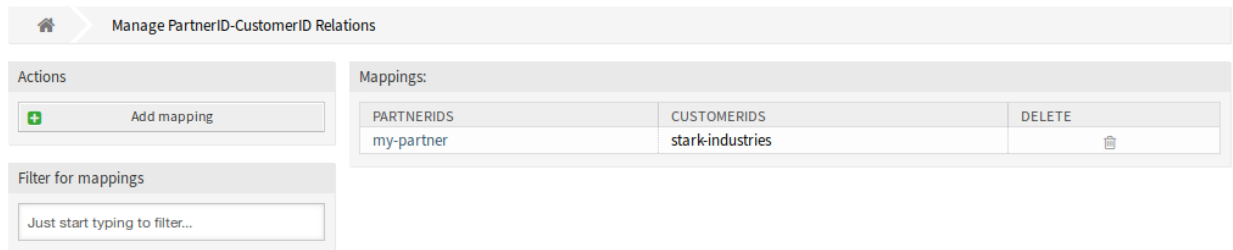


Fig. 1: Manage PartnerID-CustomerID Relations Screen

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.

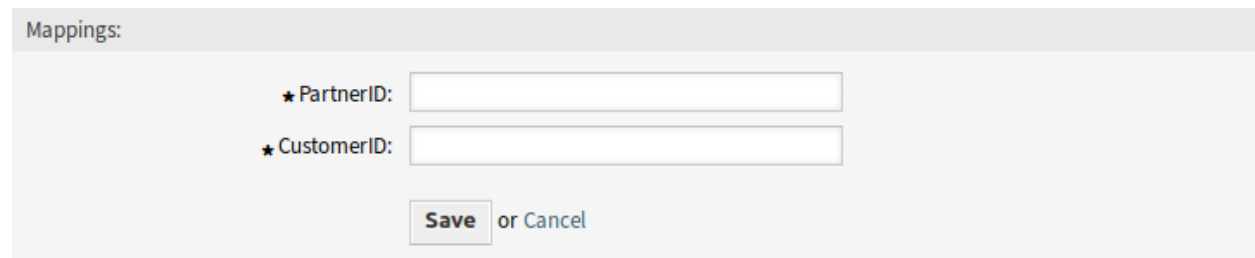
Following the relations stored in the mapping tables, access to data of certain customers is restricted for agents which have a partner ID and relations in the database. Those agents can only see customers which have customer IDs that are allowed in the database.

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 web interface for adding a mapping. At the top, there is a header 'Mappings:'. Below it, there are two input fields. The first field is labeled '★ PartnerID:' and the second is '★ CustomerID:'. Below the second field, there is a 'Save' button and the text 'or 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 *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:

PartnerID: my-partner

* CustomerID:

stark-industries

or or

Fig. 3: Edit Mapping Screen

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.

45.2 Agent Interface

This package has no agent interface.

45.3 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

46.1 Administrator Interface

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

46.1.1 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.

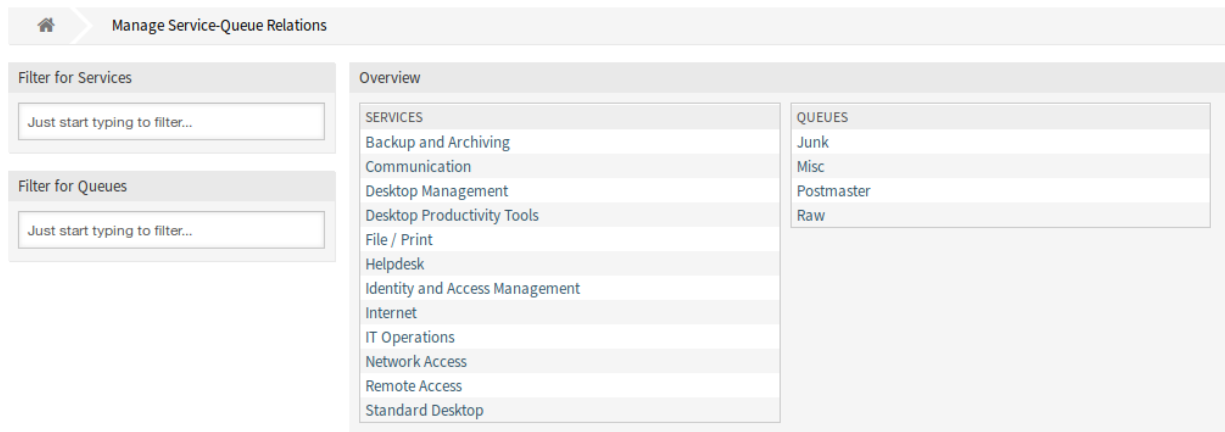


Fig. 1: Manage Service-Queue Relations Screen

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>
--

46.2 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 *Ticket Move* screen. 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 *Get All* 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.

46.3 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 *Get All* 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

47.1 Administrator Interface

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

47.1.1 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.

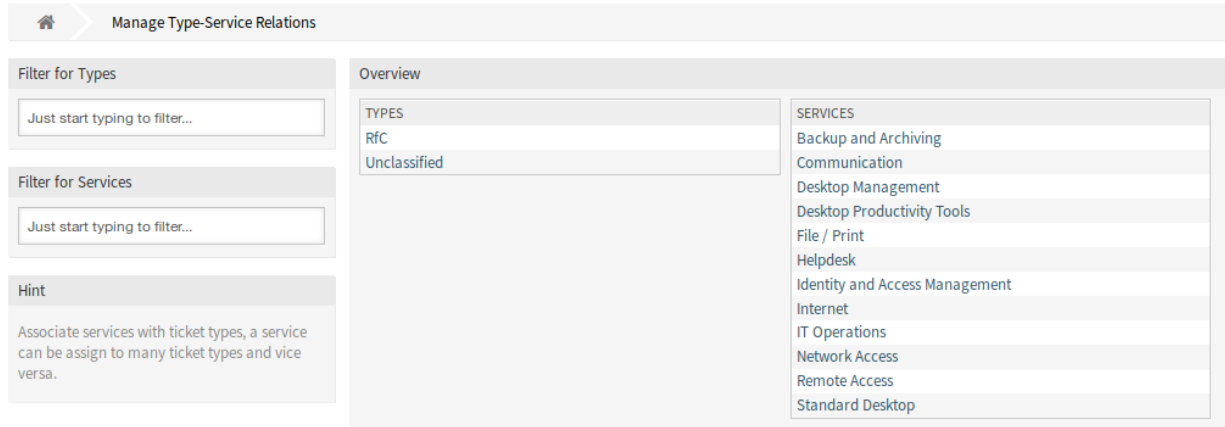


Fig. 1: Manage Type-Service Relations Screen

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.

47.2 Agent Interface

This package has no agent interface, but it extends the standard behavior of the service selection method.

47.2.1 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*.

47.3 External Interface

This package has no external interface.

Service Responsible

Typically, an inbound ticket is first assigned to an edit group/team, known as a queue, by an agent. This agent usually has no knowledge of which employee on the respective team should handle the request. The ticket is then reopened by the team that manages the queue, and it is assigned to the person who becomes responsible for taking care of the request. This means that the ticket is dispatched two times, so to speak.

This feature add-on allows you to configure your system so that specific agents are associated with each service option. This way, the agent who originally receives the inbound ticket does not need to know which employee is responsible for which services, but instead chooses only the service that needs to be provided. The assignment to a person is automatically handled by the system. Now, ticket assignment only needs to happen one time.

This functionality is, for example, a big advantage when working with a paged first-level support group. In this situation, agents who handle inbound tickets do not usually know the individual employees and their skill sets. This can cause delays in correctly assigning the ticket.

Benefits

- Tickets are assigned directly to the person responsible.
- Ticket allocation is faster and more efficient.
- Assignment to outsourced first-level support is seamless.

Target Groups

- Large companies
- Companies with a high volume of tickets
- Companies with many employees
- Service-oriented workgroups
- Companies with outsourced first-level support

Available in Service Package

- TITANIUM

Package Name in OTRS Package Manager

- OTRSServiceResponsible

48.1 Administrator Interface

This package has no administrator interface, but adds a new entry to the recipient list of ticket notifications.

48.1.1 Communication & Notifications

This feature add-on adds a notification recipient for agents, who set the services in the *Personal Preferences* as responsible services.

Ticket Notifications

After installation of the package a new option will be available for the *Send to* field of the *Recipients* widget.

Manage Ticket Notifications

This feature add-on adds a new notification option for the service responsible agent.

Ticket Notification Settings

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

See also:

For an example, see a default ticket notification which is included in a fresh **OTRS** installation.

Recipients

▼ Recipients

Send to:

Send to these agents:

Send to all group members (agents only):

Send to all role members:

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

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

Fig. 1: Ticket Notification Settings - Recipients

Send to The service responsibilities of all assigned agents can be chosen with the option *All agents responsible for the service* to notify the responsible agents.

All other setting and options are part of the **OTRS** framework.

48.2 Agent Interface

This package adds a new preferences module to assign agents as responsible for services. Each agent can be assigned as responsible to multiple services and each service can have assigned multiple agents.

48.2.1 Personal Menu

With this feature add-on each agent can be assigned to services within the related user preferences.

Personal Preferences

After installation of the package a new widget will be available in the *Notification Settings* module in the *Personal Preferences* screen.

Notification Settings

These settings are for customizing personal notifications.

Service Responsible

In this widget the agents can set services to be a responsible agent to those services.

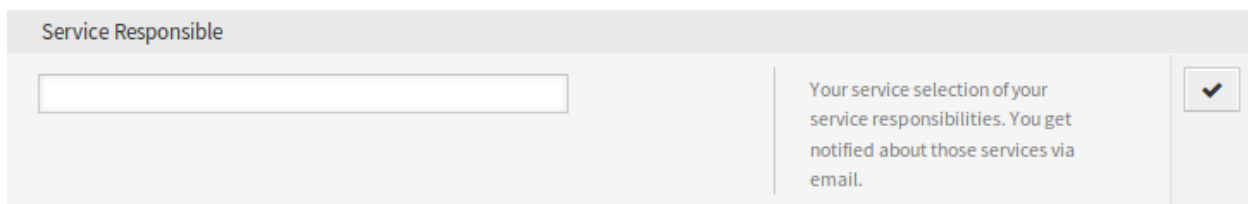


Fig. 2: Service Responsible Widget

Select one or more services from the drop-down list.

Note: To activate this widget in personal preferences, you need to enable the following system configuration options:

- `Ticket::Service`
- `Ticket::Responsible`

48.3 External Interface

This package has no external interface.

Specific Ticket Notifications

As part of standard functionality, agents with administrator permissions can create automated notifications to be sent to a certain group of recipients of agents and/or customers. These are standard notifications, e.g. for when the owner of the ticket has changed or the ticket has been moved to another queue.

Now, with the feature add-on, agents without administrator privileges can also create individual, event-based notifications, albeit with the limitation that they can only be sent to their own email address. For example, an IT-service employee can configure individual notifications in order to be informed as soon as there is a change in the priority of a specific customer's ticket.

Benefits

- Stay informed regarding ticket changes, even when working outside of the agent interface.
- React to changes more quickly.
- More personalized configuration of notifications, avoiding a flood of information.

Target Groups

- Internal and external IT
- Customer service
- Sales
- Finance

Available in Service Package

- GOLD

Package Name in OTRS Package Manager

- OTRSSpecificTicketNotifications

49.1 Administrator Interface

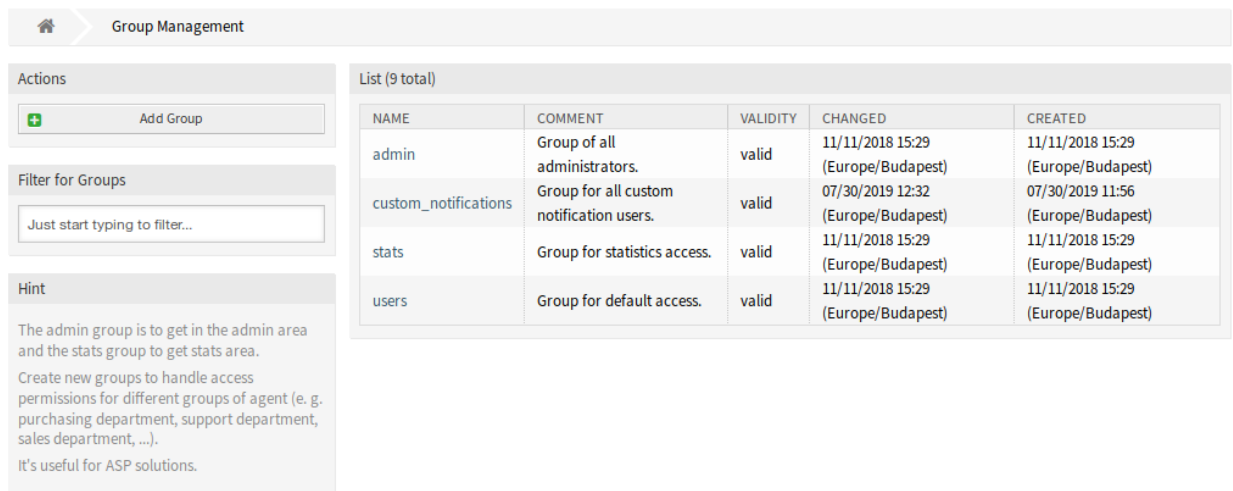
This package has no administrator interface, but extends *Ticket Notifications* screen with new buttons and adds a new group to the system.

49.1.1 Users, Groups & Roles

After installation of the package a new group is added to the system.

Groups

Access to the personal notification module is managed on a role-based access concept. After installation of the package a new group is added to the system. The group management screen is available in the *Groups* module of the *Users, Groups & Roles* group.



NAME	COMMENT	VALIDITY	CHANGED	CREATED
admin	Group of all administrators.	valid	11/11/2018 15:29 (Europe/Budapest)	11/11/2018 15:29 (Europe/Budapest)
custom_notifications	Group for all custom notification users.	valid	07/30/2019 12:32 (Europe/Budapest)	07/30/2019 11:56 (Europe/Budapest)
stats	Group for statistics access.	valid	11/11/2018 15:29 (Europe/Budapest)	11/11/2018 15:29 (Europe/Budapest)
users	Group for default access.	valid	11/11/2018 15:29 (Europe/Budapest)	11/11/2018 15:29 (Europe/Budapest)

Fig. 1: Group Management Screen

New Group

After installation of the package the following group is added to the system:

custom_notifications Every agent who is a member of this group gets a *Custom Notifications* widget in the bottom of the *Notification Settings* module of *Personal Preferences* screen.

Note: The primary administrator user (`root@localhost`) is added to all groups with permission *rw* by default.

See also:

To set the correct permissions for other agents, check the following relations:

- *Agents Groups*
- *Roles Groups*

49.1.2 Communication & Notifications

After installation of the package a new functionality will be available in the *Ticket Notifications* screen of the *Communication & Notifications* group in the administrator interface.

Ticket Notifications

This screen gets two new buttons in the left sidebar and a new column in the overview table.

NAME	COMMENT	VALIDITY	CHANGED	CREATED	PERSONAL	EXPORT	COPY	DELETE
Customer ticket article notification		valid	11/11/2018 15:29 (Europe/Budapest)	11/11/2018 15:29 (Europe/Budapest)				
Customer ticket create notification		valid	05/30/2019 15:44 (Europe/Budapest)	11/11/2018 15:29 (Europe/Budapest)				
Customer ticket state update (closed)		valid	11/11/2018 15:29 (Europe/Budapest)	11/11/2018 15:29 (Europe/Budapest)				
Customer ticket state update (other than closed)		valid	11/11/2018 15:29 (Europe/Budapest)	11/11/2018 15:29 (Europe/Budapest)				
Supervisor	To view newly created a...	valid	07/30/2019 13:47 (Europe/Budapest)	07/30/2019 13:47 (Europe/Budapest)	Yes			
Ticket create notification		valid	11/11/2018 15:29 (Europe/Budapest)	11/11/2018 15:29 (Europe/Budapest)				

Fig. 2: Ticket Notification Management Screen

Administrators are still able to see all available notification entries. To be able to differentiate between entries of administrators and agents, there is a new column *Personal* in the notification overview. The entries of agents are marked in orange and have the value *Yes* in the column *Personal*.

Furthermore it is still possible to add other agents as email recipients in every new or modified notification entry, including notification entries of agents.

Manage Personal Ticket Notifications

To add a personal ticket notification:

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

To edit a personal ticket notification:

1. Click on a ticket notification in the list of ticket notifications in a row that is displayed in orange.

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

To delete a personal ticket notification:

1. Click on the trash icon in the list of ticket notifications in a row that is displayed in orange.
2. Click on the *Confirm* button.

To list only the personal notifications:

1. Click on the *List my personal notifications only* button in the left sidebar.
2. Click on the *List all notifications* button to restore the overview.

Adding a notification can be done with the same procedure that is described in the [administrator manual](#). There is only one difference: the *Recipients* section.



Fig. 3: Ticket Notification Settings - Recipients

In case of adding a personal ticket notification, only the personal email address of the administrator or agent can be selected in the *Recipients* section.

49.2 Agent Interface

This package allows agents to set personal ticket notifications, like administrators can do.

49.2.1 Personal Menu

After installation of the package it is possible to create own notifications, like administrators can do, except for the recipient addresses. Every event which is created by an agent will send emails to the agents own primary email only.

Personal Preferences

This package adds a new *Custom Notifications* widget to the bottom of the *Notification Settings* module of *Personal Settings*.

Notification Settings

Every agent who is a member of the group `custom_notifications` gets a *Custom Notifications* widget in the bottom of the *Notification Settings* module of *Personal Settings*.

Custom Notifications

This widget contains a link that points to the *Ticket Notifications* screen of the administrator interface, if an administrator clicks on the link. If an agent clicks on it, it opens a reduced variant of the *Ticket Notifications* screen.

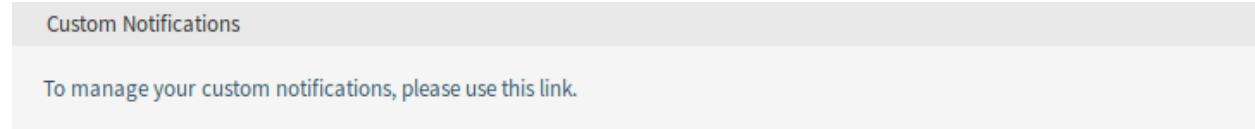


Fig. 4: Custom Notifications Widget

The *Ticket Notifications* screen does not contain the administrative elements, if an agent opens it.



Fig. 5: Ticket Notification Management Screen

To add a personal ticket notification:

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

To edit a personal ticket notification:

1. Click on a ticket notification in the list of ticket notifications in a row that is displayed in orange.
2. Modify the fields.
3. Click on the *Save* or *Save and finish* button.

To delete a personal ticket notification:

1. Click on the trash icon in the list of ticket notifications in a row that is displayed in orange.
2. Click on the *Confirm* button.

The agent is able to create his own notifications, like administrators can do, except the recipient addresses. Every event that an agent creates, will send emails to its own primary email. This recipient address is mandatory and cannot be switched to another one.

The related agent is only able to see, create, edit and/or delete his own notification entries. It is not possible to see or modify notification entries of other agents.

49.3 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

50.1 Administrator Interface


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

50.1.1 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...

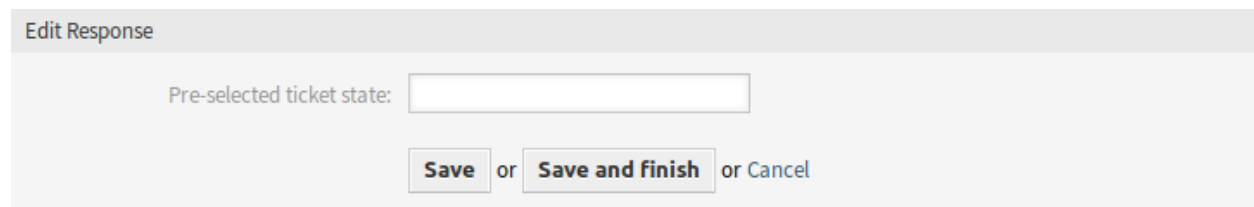
TEMPLATE NAME	PRE-SELECTED TICKET STATE
empty answer	-

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 *Ticket Compose* screen when answering a ticket using this response template.

50.2 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 Zoom* screen and click on the *Reply* menu item in the article menu.
2. Select a response template from the drop-down list.
3. The pre-selected ticket state should be set on the *Next ticket state* field in the *Ticket Compose* window.

See also:

The ticket state can be assigned in the [State Pre-selection for Templates](#) management screen.

50.3 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

51.1 Administrator Interface

This package adds two new screens to see the history of the system configuration settings.

51.1.1 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.

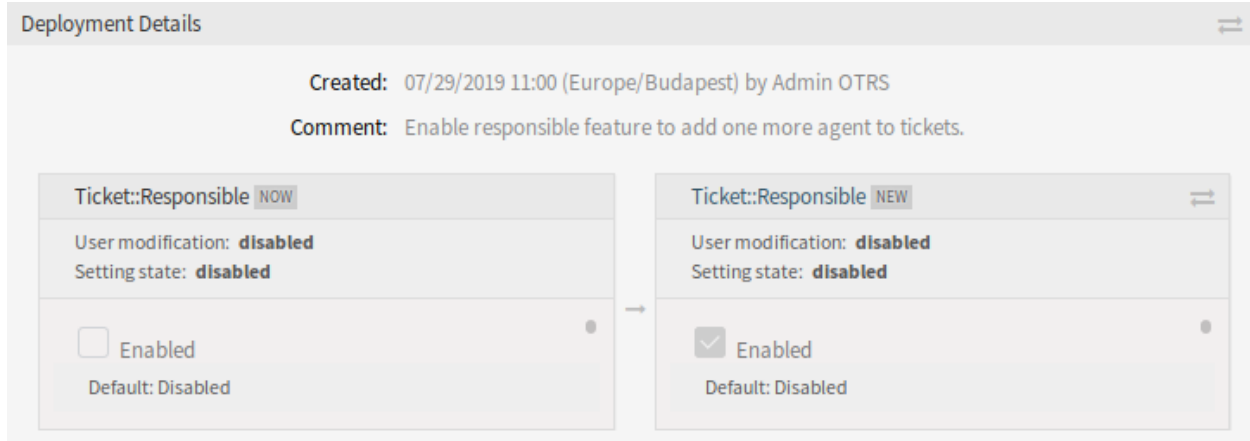


Fig. 3: Deployment Details Difference

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.

51.2 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

51.3 External Interface

This package has no external interface.

Tag Cloud

This feature add-on provides a better overview of trends in ticket information with the help of assigned tags. Where the standard version of **OTRS** offers similar functionality with a lot more clicks, the tag cloud shows the information instantly, at a single glance, and in comparison to other results. For example, it is possible to visualize the workload of single agents or departments dynamically and to see the most up-to-date results. Likewise, the frequency of incidents with certain products can be measured constantly and can be compared with each other, which leads to a faster reaction on deviations.

The following ticket attributes can be visualized in the tag cloud:

- Ticket state
- Queue
- Owner
- Responsible
- Priority
- SLA
- Type
- All dynamic fields (e.g. error classes, products, etc.)

The larger a tag cloud is, the more the defined ticket attributes occur. The numbers in the brackets show the exact number of found tickets. With a direct click on a tag, the ticket search starts automatically and the related tickets are shown instantly.

Benefits

- Find out about trends & habits at a glance.
- See the dynamic and current workload of your agents.
- React fast to anomalies.

Target Groups

- Marketing

- Human resources
- Sales
- IT service management
- And many more

Available in Service Package

- SILVER

Package Name in OTRS Package Manager

- OTRSTagCloud

52.1 Administrator Interface

This package has no administrator interface.

52.2 Agent Interface

This package includes the functionality of a tag cloud, which can be displayed as a small and/or large widget in the dashboard.

52.2.1 Dashboard

This package adds a small and/or a large tag cloud widget to the agent dashboard. The tag cloud displays a configurable number of top hits for a selectable ticket attribute or dynamic field.

Dashboard

This package adds a small tag cloud widget to the right sidebar of the dashboard and a large tag cloud widget to the content widgets of the agent dashboard. Each logged in agent has the option to configure or hide the tag cloud widgets according to personal preferences.

The tag cloud widget displays a selection of top entries of the configured source. When clicking on one of the displayed tags, a ticket request with the selected value is started as a search criterion in a new browser window.

To turn on or off the tag cloud widgets:

1. Click on the *Settings* widget in the right sidebar to expand it.
2. Select or deselect the widgets *OTRS TagCloud – Small* and *OTRS TagCloud – Large*.
3. Click on the *Save settings* button.

See also:

You can also remove a widget, if you hover the mouse over the widget header and click on the × on the top right corner to close the widget.

To configure a tag cloud widget:

1. Hover the mouse over the widget header.

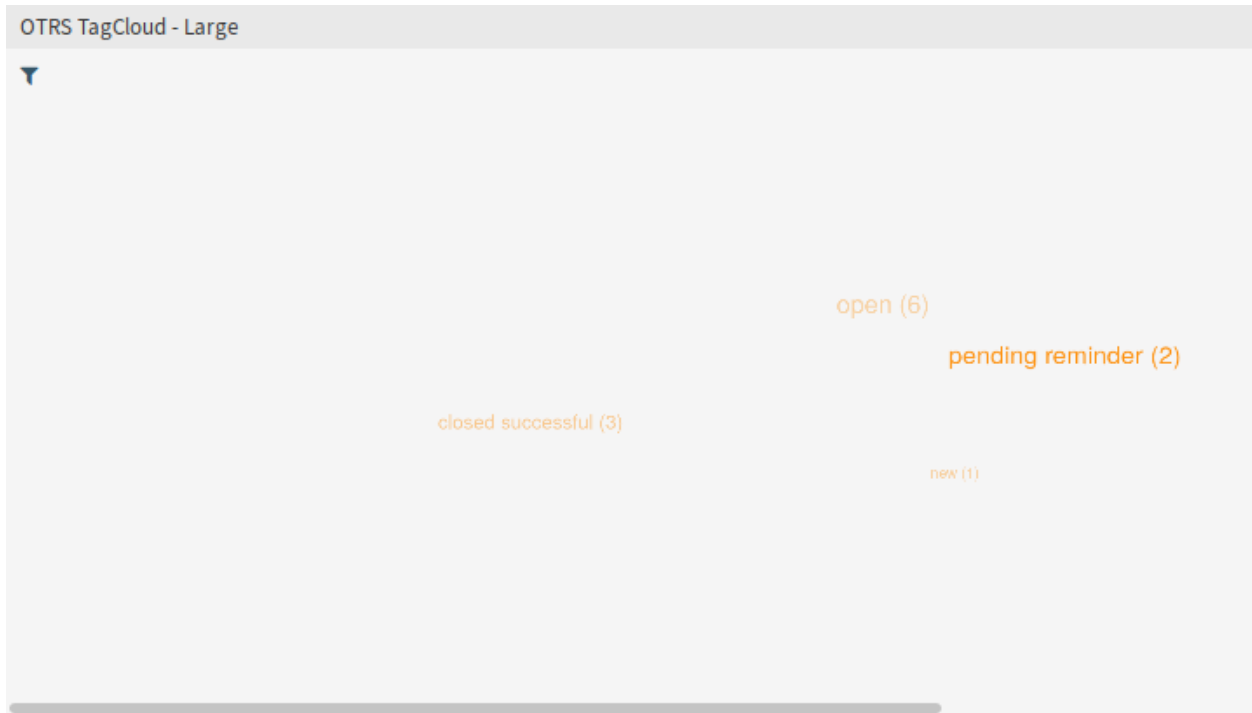


Fig. 1: Large Tag Cloud Widget

2. Click on the gear icon in the top right corner.
3. Modify the fields.
4. Click on the *Save Changes* button.

The screenshot shows the settings panel for the "OTRS TagCloud - Large" widget. It contains three configuration fields: "Shown" with a value of "50", "Source" with a value of "Queue", and "Scrollwheel zoom" with a value of "Enable". At the bottom of the panel, there are two buttons: "Save Changes" and "Cancel", separated by the word "or".

Fig. 2: Tag Cloud Settings

Each agent can configure individual tag clouds independently from other agents or tag clouds in the dashboard. The settings are stored separately for each tag cloud and agent.

Shown Select, how many tags should be displayed in the widget. The most common tags are always displayed.

Source Select, from which source should be the tags visualizing. It is possible to select between the ticket fields like priority, queue, SLA, service, status and type, as well as for ticket search configured dynamic fields.

See also:

To configure the dynamic fields for the ticket search screen, see the setting `Ticket::Frontend::AgentTicketTicketSearch###DynamicField`.

Scrollwheel zoom If enabled, the scroll wheel of the mouse can be used to zoom in or out in the tag cloud instead of scrolling the page. The zoom can be helpful if you want to enter deeper into the tag cloud.

It is possible that empty values are stored in dynamic fields. These are represented by the string *Empty*. If no data is available for the selected source, an error message with corresponding text is displayed. This tells the agent the missing data and asks him to choose another source for the tags.

The tag cloud offers the possibility to filter the tag search for queues, states, owners and responsible person. For this, the configuration parameter `FilterEnabled` must be activated in the setting of the tag cloud in the system configuration.

To filter the sources:

1. Click on the filter icon in the top left corner of the content area in the tag cloud widget.
2. Add some filter criteria with the filter options.
3. Click on the *Submit* button.

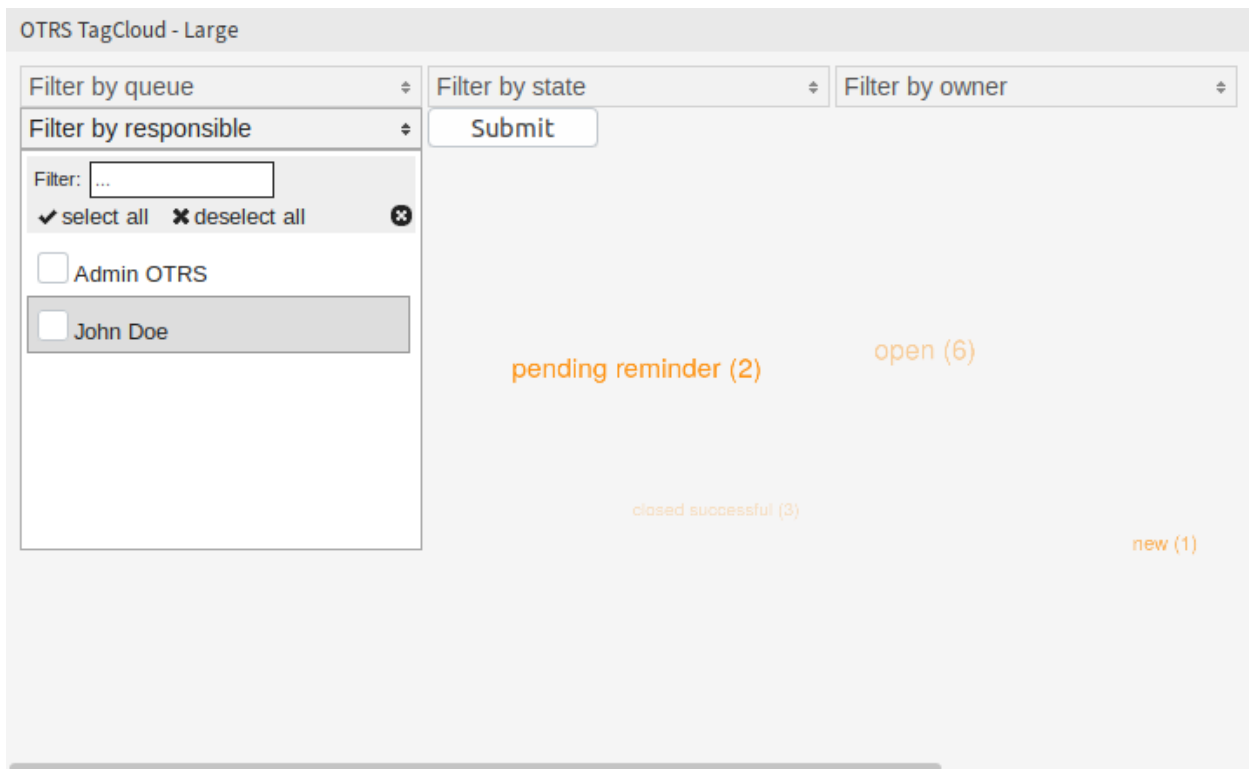


Fig. 3: Tag Cloud Filters

An asynchronous request is sent to the server that delivers the new tag data without reloading the entire page. While this request is running, an icon appears in the upper right corner of the widget.

The used filters are retained beyond the existing session and do not expire when logging off.

52.3 External Interface

This package has no external interface.

Ticket Allocation

Service organizations that need to handle a high number of tickets in a short time span and call centers that want to assign tickets automatically to staff members will both be excited about this feature add-on. New tickets are automatically assigned to agents with the smallest number of tickets to work on or to agents with suitable skills. Your service team will efficiently operate at full capacity and will be able to answer more tickets without feeling overburdened. This feature add-on turns **OTRS** into a call center software.

In the generic agent job configuration you can define the event that will trigger the ticket allocation. For example, the trigger could be the creation of a ticket, the reaching of a reminder time, or the closing of a ticket. Moreover, you can decide if only the ticket that triggered the event is allocated or if all relevant tickets are allocated to a specific agent.

You can also define which tickets are relevant. To avoid flooding your agents with tickets, you can restrict the number of allocated tickets per agent. Ticket ordering can also be configured with an additional *order module*. By default, it is the ticket creation time, but you can also choose queue priority or service times. Additionally, a maximum number of allocated tickets per queue can be defined.

Last but not least, you can also create competence groups in order to react more quickly to emergency tickets or to assign tickets to a team with special knowledge of tricky issues right from the start. By default the queue, priority, type, and group competences are active; however, you can also add SLA and service in the system configuration.

Benefits

- Optimal capacity utilization of your service team by automatic allocation of tickets.
- Simple handling of an increased number of tickets.

Target Groups

- IT service
- Sales
- Facility management
- Internal IT
- And many more

Available in Service Package

- GOLD

Package Name in OTRS Package Manager

- OTRSTicketAllocation

Note: Not compatible with the following feature add-on:

- *Advanced Generic Agent*
-

53.1 Administrator Interface

The ticket allocation is the core feature of this package. It will automatically set an owner and a configurable lock for tickets. The events that trigger an allocation check can be defined in the *Events* widget of the *Generic Agent* job management screen. Each event can be configured to allocate only the triggering ticket or all relevant tickets. Which tickets are relevant for ticket allocation can be defined by setting the ticket filters in the generic agent job.

The possible owners of a ticket have to be in the configured group of the ticket queue. Agents can be excluded from the allocation by being member in one of the configured blacklist groups. The allocation can be influenced by the competence feature (if enabled).

The competences will get summed for each agent. The group-competences will be mapped to the queue group of the ticket, and all other competences to its corresponding ticket attribute (queue, priority etc.). If all/some agents have the same competence score or the competences feature is disabled, the agent with the least amount of tickets will be set as the new ticket owner. The competences can be set for each *Agents* individually.

It is possible to define a maximum number of tickets per agent via the system configuration to prevent flooding the agent with tickets. If the configured value is reached for an agent no new ticket will get allocated to her/him. It is possible to loose this rule for tickets that get manually allocated to an agent via the system configuration, too.

By default only on-line agents will get tickets allocated. This behavior can also be changed by a system configuration setting.

53.1.1 Processes & Automation

After installation of the package a new generic agent job will be added to the system.

Generic Agent

The ticket allocation functionality is based on the event based ticket actions feature from the generic agent. The job can be executed for all matching tickets or just the ticket that has triggered the event. Additionally it is possible to define custom order modules and pre action modules for the generic agent job. This package provides one order and pre action module to allocate tickets. It uses the triggers to execute the allocation process.

After installation of the package a generic agent job called *TicketAllocation* is added to the system. This job is a basic ticket allocation configuration that can be customized and extended.

The filtering of tickets has not changed from the default generic agent behavior. For the most cases the default event configuration of the ticket allocation job did not have to be changed. The events should be self-explanatory. They represent the different ticket actions and the login/logoff of agents.

By default the job is executed via the OTRS daemon to avoid blocking the agent and preventing infinite allocation loops. Additionally it is necessary to set the prevent infinite loops to Yes.

By default the *TicketAllocation* pre action module will set the ticket lock. This will trigger another *TicketLockUpdate* event, which will start a new job and so on. The *TicketLockUpdate* event is required to make it possible to react on automatic and manual ticket unlocks to reallocate tickets. It is recommended to keep these settings.

▼ Events

Event Triggers:

TYPE	EVENT	ALL TICKETS	DELETE
Ticket	TicketCreate	<input type="checkbox"/>	
Ticket	TicketQueueUpdate	<input type="checkbox"/>	
Session	SessionCreate	<input type="checkbox"/>	
Session	SessionRemove	<input type="checkbox"/>	
Ticket	TicketStateUpdate	<input type="checkbox"/>	
Ticket	TicketLockUpdate	<input type="checkbox"/>	

Additionally or alternatively to a periodic execution, you can define ticket events that will trigger this job. The checkbox defines whether the job should search for all matching tickets or just take the one that has triggered the event.

Add Event Trigger:

To add a new event select the event object and event name.

Event execution:

Defines how the job should get executed. Please refer to the package documentation for more information. Attention: Misconfiguration can lead to endless loops.

Prevent infinite loops:

Defines whether the job can get executed in another event or not. Attention: Misconfiguration can lead to endless loops.

Fig. 1: Events Section

In the *Execute Custom Modules* widget are two select fields which define if an *Order Module* and/or *Pre Action Module* should get executed. This package adds one order module and pre action module to the system.

The order module is not mandatory to use for the allocation functionality. There are a couple of system

Execute Custom Modules

Order module:
 Use a module for custom ordering of the tickets that matched the filter criteria.

Pre action module:
 Use a module for condition based setting of ticket actions. Attention: Ticket actions may be overwritten by this module.

Execute Custom Module:

Param 1 key: Param 1 value:

Param 2 key: Param 2 value:

Param 3 key: Param 3 value:

Param 4 key: Param 4 value:

Param 5 key: Param 5 value:

Param 6 key: Param 6 value:

Fig. 2: Execute Custom Modules Section

configuration settings that can influence the order of the tickets. If the order should get manipulated it is necessary to use the *TicketAllocation* order module. The pre action module is necessary to manipulate the ticket attributes that the tickets should get set. There are some system configuration settings to influence the allocation process, too.

It is possible to set additional attributes the processed tickets should get set. The *TicketAllocation* pre action module will set the owner and the lock (if configured) for the tickets, if there are no errors in the allocation process and a matching agent can be found. If no allocation can be made, it is possible to set default values in the *Update/Add Ticket Attributes* widget of the generic agent for the owner and lock. If an allocation is made these values will be overwritten.

53.1.2 Users, Groups & Roles

This package adds a new configuration screen to the agent preferences screen, which is accessible from the *Agent Management* screen of the administrator interface.

Agents

With this package it is possible to define different competences for agents.

The competences can be enabled and disabled via the system configuration. By default the ticket competences group, priority, queue and type are enabled. Role, SLA and service are disabled but can be enabled via the system configuration.

To set the competences for an agent:

1. Open the *Agents Management* screen in the administrator interface.

2. Select an agent from the list.
3. Click on the *Edit personal preferences for this agent* button in the left sidebar.
4. Click on the *Competences* module in the agent personal preferences.

Group competence(s)	
"Group" competence(s): <input type="text" value="admin"/> <input type="button" value="v"/> <input type="text" value="1 very low"/> <input type="button" value="x"/>	Define group competences for this user. <input type="button" value="v"/>
<input type="text" value="users"/> <input type="button" value="v"/> <input type="text" value="5 very high"/> <input type="button" value="x"/>	
<input type="button" value="⊕"/>	
Priority competence(s)	
"Priority" competence(s): <input type="button" value="⊕"/>	Define priority competences for this user. <input type="button" value="v"/>
Queue competence(s)	
"Queue" competence(s): <input type="text" value="Misc"/> <input type="button" value="v"/> <input type="text" value="5 very high"/> <input type="button" value="x"/>	Define queue competences for this user. <input type="button" value="v"/>
<input type="button" value="⊕"/>	
Type competence(s)	
"Type" competence(s): <input type="button" value="⊕"/>	Define type competences for this user. <input type="button" value="v"/>

Fig. 3: Competences Screen

The competences have no direct influence on the behavior of the system. They can be used in other functionalities to make decisions based on the configured values and influence the behavior.

By default the relevant tickets will be allocated ordered by their internal ticket ID and their creation time. This package provides an order module for changing the order of the relevant tickets, based on different system configuration settings. The ticket order can be influenced by configurable queue priorities and service times.

Additionally it is possible to set a maximum number of tickets per queue in a row that should get allocated. If the limit is reached, the next different queue will be preferred. After that, the counter for the queue is reset.

See also:

Take a look at the configuration options in the system configuration for this package.

53.2 Agent Interface

This package has no agent interface.

53.3 External Interface

This package has no external interface.

Ticket Forms

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 *Ticket Pending* screen)

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:

- *Advanced Editor*
 - *Categories for Text Modules*
 - *Clis in Customer Frontend*
 - *Dynamic Field Cl*
 - *Hide/Show Dynamic Fields*
 - *Restore Pending Information*
 - *Service-based Queue Routing*
-

54.1 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.

Ticket Forms itself, enables to set a custom maximum length for services and SLA names in ticket creation screens.

This feature can work cooperatively with *Ticket Workflow* package by integrating defined ticket workflows into a template and trigger them automatically when the ticket is created.

54.1.1 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 input field. Below the filter is a hint section explaining that dynamic ticket templates are default text and dynamic fields to help agents write faster tickets. 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)	

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.

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

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.

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.

Workflow Select a ticket workflow, that has been added to the system.

Note: This field is displayed only, if the *Ticket Workflow* package is installed.

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 stems for this resource at the bottom of both add and edit screens.

Edit Template

General:

★ Name:

Comment:

★ Valid:

★ Frontend:

Restrictions:

★ Type:

★ Service:

Template:

Workflow:

Subject:

Body:

B *I* U ~~S~~ |

Format ▾ | Font ▾ | Size ▾ |

Oil Change
Oil Filter Change
Air Filter Change
Fluids Check

Attachments:

Brand:

VW Model:

VW Production Facility:

Peugeot Model:

Peugeot Production Facility:

Fuel:

Accessories:

Remarks:

Registration Date:

Invoice Date:

or
 or




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

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
```

54.1.2 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.

54.1.3 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.

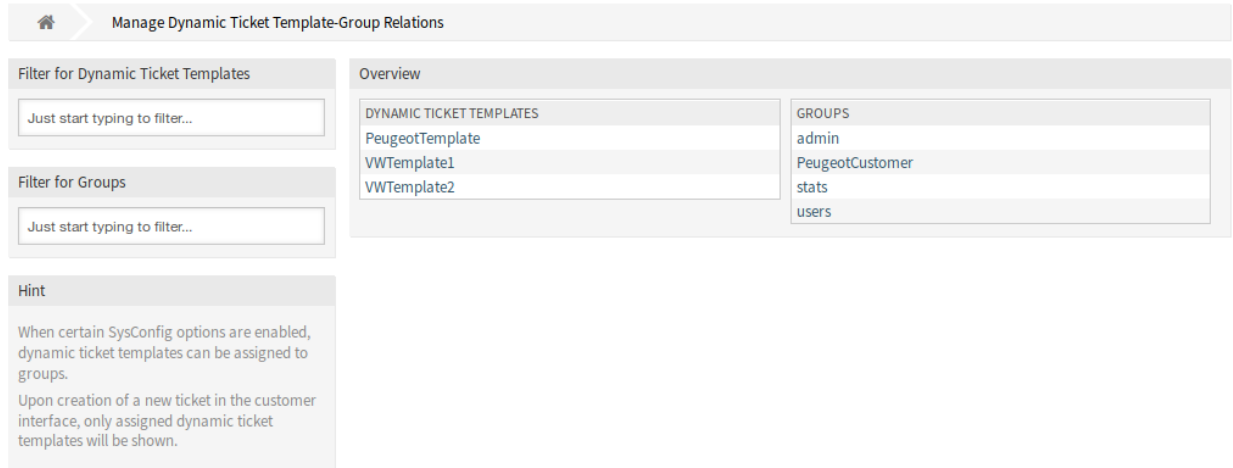


Fig. 6: Manage Dynamic Ticket Template-Group Relations

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.

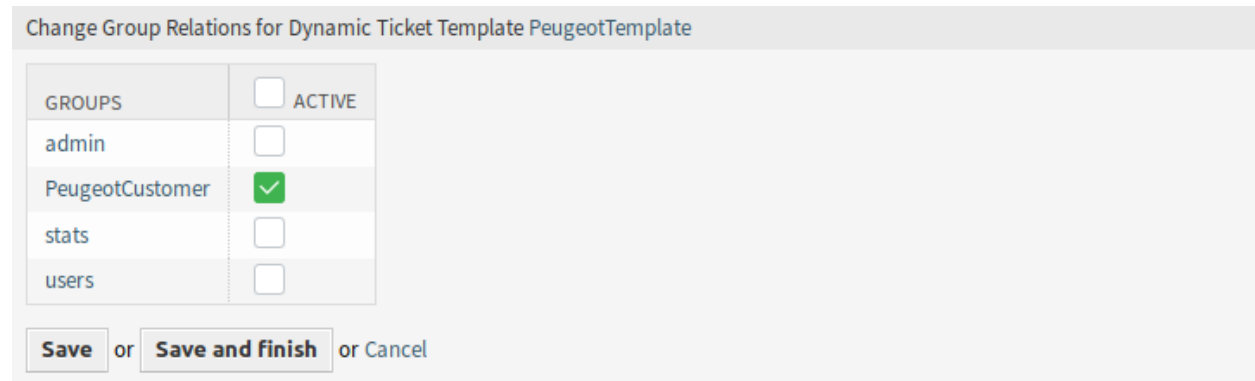


Fig. 7: Change Group Relations for Dynamic Ticket Template

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.

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"/>

or or

Fig. 8: Change Dynamic Ticket Template Relations for Group

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.

Warning: Accessing a dynamic ticket template or a group provides no back link to the relations screen.

54.1.4 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 for *New Phone Ticket* in the system configuration setting `Ticket::Frontend::AgentTicketPhone###DynamicField` or disable the dynamic fields:

```
Brand → 0 - Disabled
VWModel → 0 - Disabled
VWProductionFacility → 0 - Disabled
PeugeotModel → 0 - Disabled
PeugeotProductionFacility → 0 - Disabled
Fuel → 0 - Disabled
Accessories → 0 - Disabled
Remarks → 0 - Disabled
RegistrationDate → 0 - Disabled
InvoiceDate → 0 - Disabled
```

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 already have ITSM packages 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 ITSM 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
Comments	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
Comments	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
Comments	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 *New phone ticket* menu item of the *Ticket* menu in the navigation 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
 - To 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.

54.1.5 Dynamic Ticket Templates and Ticket Workflows

For this example we will use the *VW Template 1* defined previously.

The goal to activate a predefined ticket workflow when create a new ticket using a particular dynamic ticket template.

See also:

Please follow the *Ticket Workflow* documentation for the steps to create a new ticket workflow.

Create a ticket workflow:

- Name: VWWorkflow1
- Comment: VW workflow 1
- Valid: valid
- Clone tasks from: -

Create a workflow task:

- Name: Oil Change Task
- To queue: Misc
- Owner: -
- Depends on: -
- Description: Change the oil
- Attachment: -
- Priority: 3 normal

Create an other workflow task:

- Name: Oil Filter Change Task
- To queue: Misc
- Owner: -
- Depends on: Oil Change Task
- Description: Change the oil filter
- Attachment: -
- Priority: 3 normal

Create an other workflow task:

- Name: Air Filter Change Task
- To queue: Misc
- Owner: -
- Depends on: -
- Description: Change the air filter
- Attachment: -
- Priority: 3 normal

Create an other workflow task:

- Name: Fluids Check
- To queue: Misc
- Owner: -
- Depends on: Oil Change Task
- Description: Check all fluids
- Attachment: -
- Priority: 4 high

Note: When dynamic ticket templates detect the installation of *Ticket Workflow* package, more options are revealed in the template edit screen.

To assign a ticket workflow to a dynamic ticket template:

1. Open the *Dynamic Ticket Templates* module of the *Ticket Settings* group in the administrator interface.
2. Click on the *VW Template 1* template name to edit it.
3. Set the workflow field to *VW Workflow 1*.
4. Click on the *Save* button.

To activate a template and a workflow:

1. Click on the *New phone ticket* menu item of the *Ticket* menu in the navigation bar.
2. A new *Ticket Workflow* field has appeared. This field is just informative and can not be changed in ticket create screens, just in the dynamic ticket template definition. It shows the name of the ticket workflow to be applied automatically when the new ticket is created.
3. Fill the form with the following data:
 - Type: Unclassified
 - From customer: Customer 1
 - Service: VW Service
 - Dynamic Ticket Template: VW Template 1
4. After selecting the *VW Template 1*, the *Ticket Workflow* field will be filled with the *VW Workflow 1*.
5. Fill any other required field and submit the ticket.
6. After the ticket is created, click on the ticket number in the notification area at the top of the screen to open the new ticket.
7. Look for the link table, either in the sidebar if configuration setting is set to *simple* or at the bottom of the page if is set to *complex*.
8. Verify that the list of liked tickets are the same as the tasks that were defined in the *VW Workflow 1* ticket workflow.

54.2 Agent Interface

This package has no agent interface.

See also:

There is a detailed use-case example in *Administrator Interface* chapter.

54.3 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`

55.1 Administrator Interface

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

55.1.1 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.

55.2 Agent Interface

This package has no agent interface.

55.3 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

56.1 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**.

56.1.1 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>`.

56.2 Agent Interface

This package has no agent interface.

56.3 External Interface

This package has no external interface.

Ticket Time Unit Dropdown

For service organizations that must meet strict service level agreements (SLAs), the documentation of time units is fundamental. For this purpose, the standard version of **OTRS** provides a field without any unit that can be filled in however you wish. However, the correct documentation of the working time with this field requires a company-wide understanding and knowledge of time and billing intervals. Unfortunately, and especially in global companies with several locations, this is not always the case. This can cause working times to be documented incorrectly and be a source of confusion for customers.

The problem can be solved by using this feature add-on. It makes it possible to set time units in a drop-down list like minutes or hours.

It is possible to define the following values:

- Selectable values
- Displayed name of value (e.g. *hour* or *minute*)
- Set default value

These definitions allow you to take into account the individual time and billing charge needs of your company. For example, when a quarter of an hour has already started, it could be calculated as a 15-minute interval automatically.

Benefits

- Reduction of error ratio when entering time units.
- Unmistakable definition of intervals and default values reduces documentation time.

Target Groups

- IT service management
- Internal IT service
- Customer service/support
- Globally active companies

Available in Service Package

- SILVER

Package Name in OTRS Package Manager

- OTRSTicketTimeUnitDropdown

57.1 Administrator Interface

This package has no administrator interface.

Note: For using this feature add-on, it is necessary to activate the drop-down lists via system configuration.

To activate the drop-down lists:

1. Go to *System Configuration* screen.
2. Select *OTRSTicketTimeUnitDropdown* in the *Navigation* widget.
3. Navigate to *Frontend* → *Agent* in the navigation tree.
4. Activate the settings for the first, second and third drop-down lists, if needed.

To use a custom time unit:

1. Change the value of the `Name` key to the custom time unit name.
2. Add possible values to `Settings` key, where the first entry is the time in seconds and the second entry is the label to be displayed in the drop-down list.

It is possible to set default value for each setting using the `Default` key and any valid value from the **second** entry of the `Settings` key. This default value will be pre-selected in the drop-down list on the ticket screens.

Now open the *New Phone Ticket* screen or any other screen using time units on the *Agent Interface* and use the drop-down lists for hours and minutes to set the time units instead of an input field.

57.2 Agent Interface

This package transforms the *Time units* input field into drop-down lists. The drop-down lists can be configured in the system configuration of the *Administrator Interface*.

To use the configured time units drop-down lists:

1. Open the *New Phone Ticket* screen or any other screen using time units.
2. Scroll down to the bottom of the form and check the new *Time units* field.

57.3 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 view, 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

Note: Not compatible with the following feature add-on:

- *Advanced Escalations*
-

58.1 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.

58.1.1 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`.

58.2 Agent Interface

With this feature add-on the agents can create watch lists and they can be notified on various events.

A watch list can contain unlimited number of tickets and a ticket can be assigned to unlimited number of watch lists.

58.2.1 Tickets

This package provides ticket watch lists and reminders for them in the *Ticket Zoom* screen, as well as adds management screen to watch lists.

Ticket Zoom

Use this screen to add tickets to a watch list or to remove tickets from a watch list. The *Ticket Zoom* screen is available, if you click on a ticket in any other screens.

Ticket Menu

After installation of the package, two new menu items will be added to the ticket menu.

Watchlists The ticket can be assigned to a watch list via this menu item. Additionally, the ticket can be removed from a watch list.

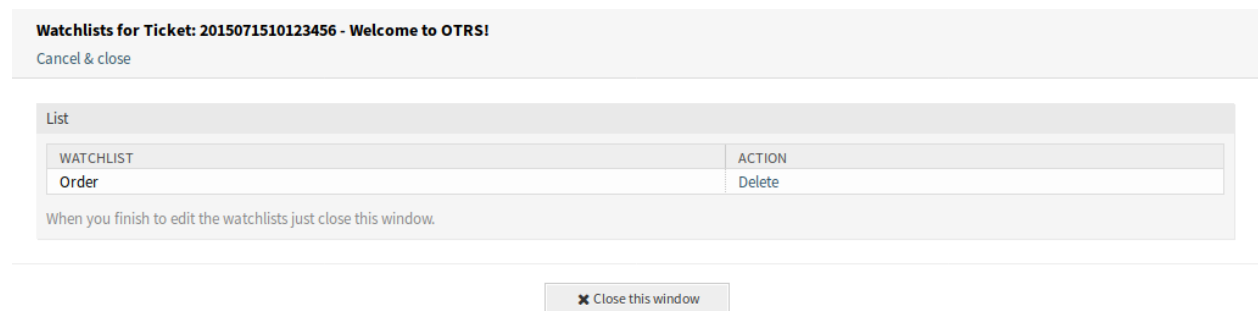


Fig. 1: Ticket Watchlists Screen

Click on the *Add* link in the *Action* column to add the ticket to the selected watch list. The *Add* link will be changed to *Delete* and the ticket will be added in the background.

Click on the *Delete* link in the *Action* column to remove the ticket from the selected watch list. The *Delete* link will be changed to *Add* and the ticket will be removed in the background.

When you finish to edit the watch lists just close this window.

Watchlist Reminder Use this menu item to set a reminder for the ticket in the watch list. For this, the agent has to be subscribed to the ticket or it has to be assigned to one of his watch lists.

Note: This menu item is visible only, if the ticket is already added to a watch list.

Reminder active Controls, whether the reminder is active or not.

Reminder Time Set the date and time when the reminder should be displayed.

Watchlist Reminder for Ticket: 2015071510123456 - Welcome to OTRS!
 Cancel & close

Options

Reminder active:

Reminder Time: / / :

Reminder for:

Note:

Fig. 2: Ticket Watchlist Reminder Screen

Reminder for Select a watch list from the drop-down list.

Note Add additional information to the reminder text.

This information will also be shown within the watch list overview.

A watch list belongs to the agent who created it. The list can be moved to another agent in the *Watchlist Management* screen. When moving a watch list to another agent, reminders will also be moved.

In this case, where the owner of the watch list 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 watch list containing the ticket and/or still is subscribed to the ticket.

There can only be one active reminder per agent and ticket. Notifications are only being sent every ten minutes.

Watchlist Management

Use this screen to manage watch lists. The watch list management screen is available in the *Watchlist Management* menu item of the *Tickets* menu or via the eye icon from the toolbar.

Watchlist Management: Overview

Actions

WATCHLIST NAME	AGENT ARTICLE NOTIFY	CUSTOMER ARTICLE NOTIFY	OWNER CHANGE NOTIFY	QUEUE CHANGE NOTIFY
Order	Yes	-	-	-

Fig. 3: Watchlist Management Screen

Manage Watch Lists

To add a watch list:

1. Click on the *Create new watchlist* button in the left sidebar.
2. Fill in the required field.
3. Click on the *Submit* button.

Watchlist data

General

★ Watchlist Name:

Notify on

New Agent Article:

New Customer Article:

New Owner:

New Queue:

New State:

Target States:

Submit or Cancel

Fig. 4: Add Watchlist Screen

To see the tickets in a watch list:

1. Click on a watch list in the list of watch lists.
2. See the tickets in the overview table.

Ticket list

TICKET#	CUSTOMERID	TITLE	QUEUE	STATE	PRIORITY	WATCHED SINCE	LAST CHANGE
2015071510123456		Welcome to OTRS!	Raw	new	3 normal	08/09/2019 14:25 (Europe/Budapest)	11/11/2018 15:29 (Europe/Budapest)

Fig. 5: Watchlist Details Screen

To edit a watch list:

1. Click on a watch list in the list of watch lists.
2. Click on the *Edit watchlist* button in the left sidebar.
3. Modify the field.
4. Click on the *Submit* button.

To assign deputies to a watch list:

Watchlist data

General

★ Watchlist Name:

Notify on

New Agent Article:

New Customer Article:

New Owner:

New Queue:

New State:

Target States:

Submit or Cancel

Fig. 6: Edit Watchlist Screen

1. Click on a watch list in the list of watch lists.
2. Click on the *Assign deputies to watchlist* button in the left sidebar.
3. Select deputy agents from the list.
4. Click on the *Submit* button.

Watchlist deputies

Deputies:

Submit or Cancel

Fig. 7: Watchlist Deputies Screen

To change the watch list owner:

1. Click on a watch list in the list of watch lists.
2. Click on the *Change watchlist owner* button in the left sidebar.
3. Select an agent from the list.
4. Click on the *Submit* button.

Warning: The original owner agent will not have access to the watch list anymore.

To export the ticket list from a watch list:

1. Click on a watch list in the list of watch lists.
2. Click on the *Export ticket list* button in the left sidebar.

Watchlist owner

Owner:

Submit or Cancel

Fig. 8: Watchlist Owner Screen

3. Select the output format.
4. Click on the *Export* button.
5. Save the file somewhere in the file system.

Watchlist export

Export format:

Export or Cancel

Fig. 9: Watchlist Export Screen

To delete a watch list:

1. Click on a watch list in the list of watch lists.
2. Click on the *Delete watchlist* button in the left sidebar.
3. Click on the *OK* button in the confirmation dialog.

Watchlist Management: Details - Order

Actions		Ticket list							
	Go to overview	TICKET#	CUSTOMERID	TITLE	QUEUE	STATE	PRIORITY	WATCHED SINCE	LAST CHANGE
	Edit watchlist	2015071510123456		Welcome to OTRS!	Raw	new	3 normal	08/09/2019 14:25 (Europe/Budapest)	11/11/2018 15:29 (Europe/Budapest)
	Assign deputies to watchlist								
	Change watchlist owner								
	Export ticket list								
	Delete watchlist								

Fig. 10: Delete Watchlist Screen

Tickets can be added to the watch list or removed from the watch list in the *Ticket Zoom* screen.

Watch List Settings

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

Watchlist 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.

New Agent Article Set to *Yes*, if the watch list owner should be notified, when an agent adds a note to a ticket added to the watch list.

New Customer Article Set to *Yes*, if the watch list owner should be notified, when a customer user creates an article in the external interface.

New Owner Set to *Yes*, if the watch list owner should be notified, when a ticket added to the watch list has a new owner.

New Queue Set to *Yes*, if the watch list owner should be notified, when a ticket added to the watch list has been moved to an other queue.

New State Set to *Yes*, if the watch list owner should be notified, when a state is changed for a ticket added to the watch list.

Target States The states selected here will activate the *New State* notification above. Notifications for status changes are only being sent if the new status matches the configured destination status.

If the *New State* notification is set to *Yes*, this field becomes mandatory.

Watch list owners and their deputies can configure the notifications separately. The configuration of the destination status is only allowed for the watch list 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 Notifications](#) module of the administrator interface.

58.3 External Interface

This package has no external interface.

Ticket Workflow

The feature add-on allows one to easily define business workflow templates for common tasks. Workflow *task* tickets can be dependent on any other task in that workflow, giving a high level of control over the order in which tasks are completed. A task that is dependent on preceding tasks will not appear in the queue until the preceding tasks are closed.

Each task ticket within a workflow can be predefined using these fields:

- Queue
- Subject
- Body
- Owner
- Priority
- Attachments

Individual tasks can be made dependent on another corresponding task within a workflow. This dependence means that tickets will only be shown in a queue when the dependent preceding ticket has changed to the status *closed*.

An agent can start a workflow through an existing ticket. The content of the original ticket will then be copied to all task tickets. All agents will have a quick overview of their tasks in their personal dashboard widgets.

Distinctive criteria:

- No further workflows can be started within an existing workflow.
- A task can be dependent on another task within a workflow.
- Workflows can only be started in the agent interface.

Benefits

- Enables the improvement of handling processes.
- Relieves agents.
- Simplifies data collection.

- Reduces error sources.

Target Groups

- IT service management
- Human resources
- Finance
- Internal IT

Available in Service Package

- SILVER

Package Name in OTRS Package Manager

- OTRSTicketWorkflow

Note: This feature add-on is also available for the OTRS::ITSM module.

59.1 Administrator Interface

After installation of the package a new ticket state and a new administrator module will be added to the system.

59.1.1 Ticket Settings

After installation of the package a new state will be added to ticket states, as well as a new module *Ticket Workflows* will be available in the *Ticket Settings* group of the administrator interface.

States

After installation of the package a new state is added to the system for the task tickets.

New State

task pending dependency Tickets with unresolved dependencies are put into this state which is a type of removed. A removed indicates that the ticket is not viewable and will not appear in the usual views. A removed does not prevent the display in the linked ticket boxes.

When the task tickets are created, the master ticket is put into the state defined as initial state in the system configuration or into *new*.

Ticket Workflows

Use this screen to add workflows and tasks to the system. The ticket workflow templates management screen is available in the *Ticket Workflows* module of the *Ticket Settings* group.

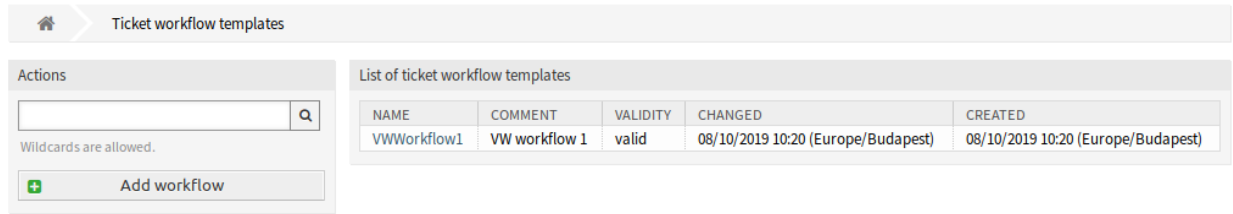


Fig. 1: Ticket Workflow Templates Screen

Manage Ticket Workflow Templates

To add a ticket workflow template:

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

Fig. 2: Add Ticket Workflow Template Screen

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

To view the details of a ticket workflow template and go to the management screen:

1. Click on a ticket workflow template in the list of ticket workflow templates.

To edit a ticket workflow template:

1. Click on a ticket workflow template in the list of ticket workflow templates.
2. Click on the *Edit workflow* button in the left sidebar.
3. Modify the fields.
4. Click on the *Save* or *Save and finish* button.

To edit the task dependencies:

1. Click on a ticket workflow template in the list of ticket workflow templates.
2. Click on the *Edit task dependencies* button in the left sidebar.

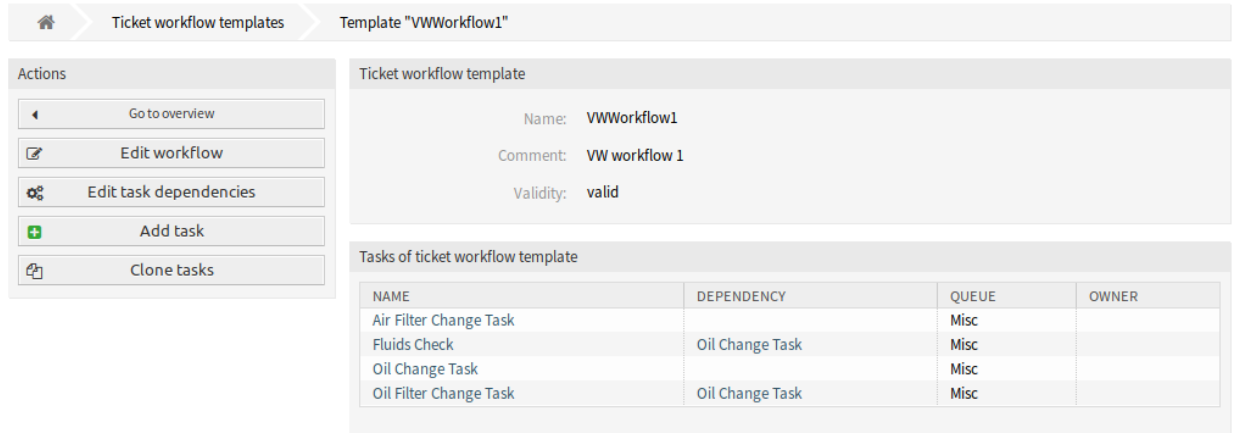


Fig. 3: Ticket Workflow Template Details Screen

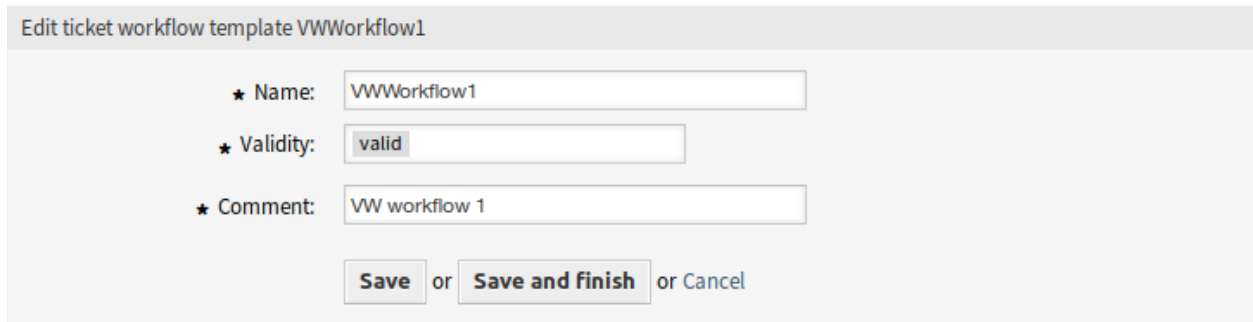


Fig. 4: Edit Ticket Workflow Template Screen

3. Add or remove task in the *Dependencies* column.
4. Click on the *Save* or *Save and finish* button.

Ticket workflow template

Name: **VVWorkflow1**

Comment: **VW workflow 1**

Validity: **valid**

Tasks of ticket workflow template

NAME	DEPENDENCY	QUEUE	OWNER
Air Filter Change Task	<input type="text"/>	Misc	
Fluids Check	Oil Change Task x	Misc	
Oil Change Task	<input type="text"/>	Misc	
Oil Filter Change Task	Oil Change Task x	Misc	

Save or **Save and finish** or Cancel

Fig. 5: Edit Ticket Workflow Task Dependencies Screen

To add a task to a ticket workflow template:

1. Click on a ticket workflow template in the list of ticket workflow templates.
2. Click on the *Add task* button in the left sidebar.
3. Fill in the required fields.
4. Click on the *Save* button.

To view the details of a task in a ticket workflow template:

1. Click on a ticket workflow template in the list of ticket workflow templates.
2. Click on a task in the list of tasks.

To edit a task in a ticket workflow template:

1. Click on a ticket workflow template in the list of ticket workflow templates.
2. Click on a task in the list of tasks.
3. Click on the *Edit task* button in the left sidebar.
4. Modify the fields.
5. Click on the *Save* or *Save and finish* button.


To delete a task from a ticket workflow template:

1. Click on a ticket workflow template in the list of ticket workflow templates.
2. Click on a task in the list of tasks.
3. Click on the *Remove task* button in the left sidebar.
4. Click on the *OK* button in the confirmation dialog.

Edit task of ticket workflow template VWWorkflow1












★ Name:


★ To queue:

Owner: 







* Depends on:

★ Description:


B *I* U ~~S~~ |   |   |     |   |  |

← → | 

Format ▾ | Font ▾ | Size ▾ |

 ▾  ▾   Source  

Change the oil

Attachments: 
Click to select files or just drop them here.

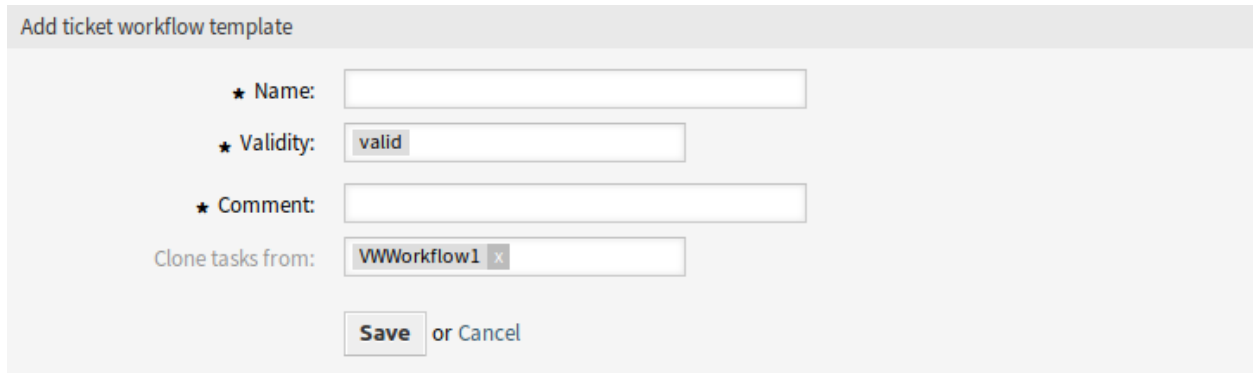
Priority:

or or

Fig. 8: Edit Ticket Workflow Task Screen

To clone a ticket workflow template:

1. Click on a ticket workflow template in the list of ticket workflow templates.
2. Click on the *Clone tasks* button in the left sidebar.
3. Fill in the required fields.
4. Click on the *Save* button.



The screenshot shows a form titled "Add ticket workflow template". It contains the following fields and controls:

- Name:** A text input field with an asterisk (*) indicating it is mandatory.
- Validity:** A dropdown menu with "valid" selected.
- Comment:** A text input field with an asterisk (*) indicating it is mandatory.
- Clone tasks from:** A dropdown menu with "VWorkflow1" selected and a close button (x).
- Buttons:** "Save" and "Cancel" buttons.

Fig. 9: Clone Ticket Workflow Template Screen

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

Ticket Workflow 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.

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

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

Clone tasks from Select an other workflow to clone its tasks from. This field is only available when using the *Clone tasks* button.

Workflow templates can inherit attributes and dynamic fields from the master ticket. Each ticket attribute or dynamic field to be copied from master ticket to task has to be defined in system configuration.

Ticket Workflow Task 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.

To queue * The queue, where the new ticket will be placed to.

Owner The owner is an agent, who actually works with the new ticket.

Depends on Select an other task from which the current task depends on. The dependency task must be finished before starting the current task.

Description Longer text description of the ticket.

Note: It is not possible to use images within the description of a task. If there is a need for images within tasks, they can be uploaded separately as attachments.

Attachments It is possible to add attachments to the ticket. Click on the field to add an attachment, or just drop some files here.

Priority New tickets have *3 normal* priority by default. Select an other priority if you want to change it.

59.2 Agent Interface

This package provides support for workflows based on master and task tickets.

59.2.1 Tickets

After installation of the package a new menu item *Workflows* will be available in the ticket menu in the *Ticket Zoom* screen.

Ticket Zoom

Use this screen to add workflows to tickets. The *Ticket Zoom* screen is available, if you click on a ticket in any other screens.

Ticket Menu

After installation of the package, a new menu item will be added to the ticket menu.

Workflow When the ticket is neither a master nor a task, then a selection of workflow templates is shown. When a template is selected and submitted, then the appropriate task tickets are generated. The generated task tickets are linked to the master ticket.

Create workflow tickets for ticket: 2019080710000017 - Test

[Cancel & close](#)

Workflow Options

★ Workflow:

Fig. 10: Ticket Workflow Screen

The task tickets are linked with *Task* link to their master ticket. The dependent task tickets are linked with *Depends On* link to the required ticket.

The deadline notice in the ticket workflow task is the solution time for the ticket. It is necessary to set an SLA escalation for this. If a defined service level agreement (SLA) is in danger of being breached, the system sends out an automatic warning message via email.

59.3 External Interface

This package has no external interface.

Time Accounting Quota

This feature add-on is perfect for call centers and IT service providers or any company that wants to assign time quotas to its customers and manage them in **OTRS**.

This feature add-on provides a new dashboard widget in the customer information center where time quotas and reference numbers, such as purchase order numbers, can be assigned to customers and where the current status of the quota is shown.

Additionally, agents can record the working time for the customer in the ticket view, and the customer can see the status of the quota in the external interface. With this feature add-on, **OTRS** can be used as a time accounting software.

Benefits

- Simple time recording and transparent documentation in one system.
- Overview of current quotas facilitates the planning and implementation.

Target Groups

- IT service management
- Call centers
- Service providers
- All companies

Available in Service Package

- SILVER

Package Name in OTRS Package Manager

- OTRSTimeAccountingQuota

60.1 Administrator Interface

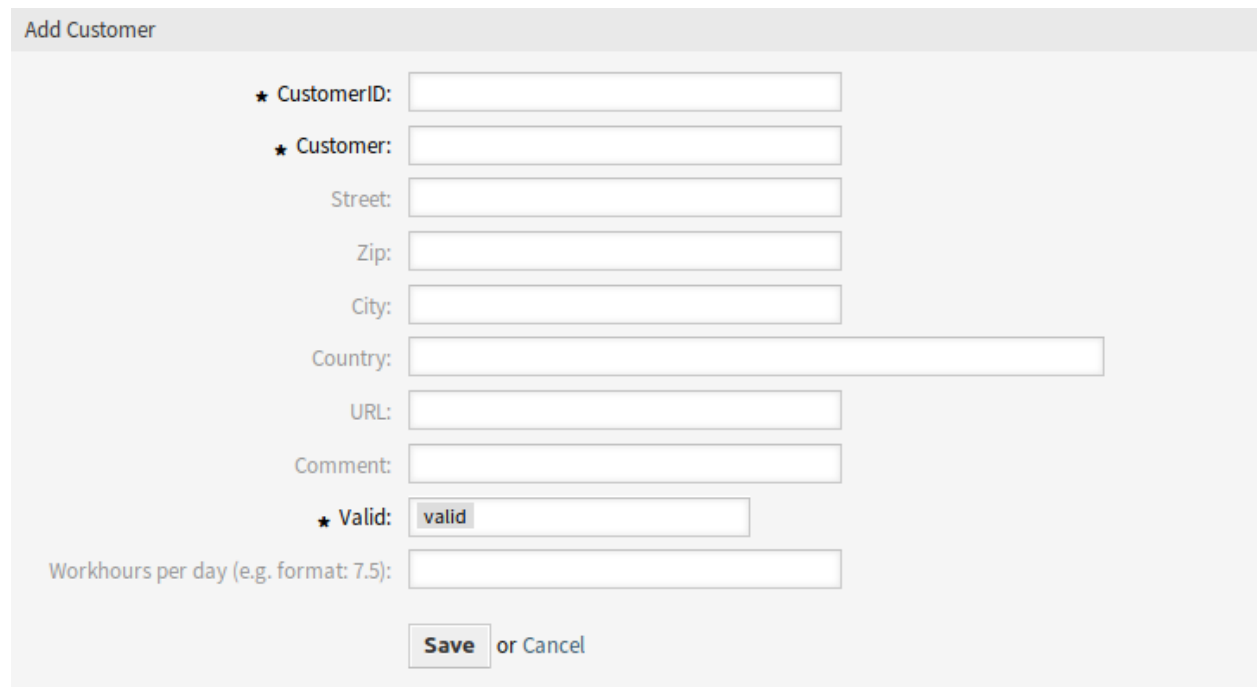
This module adds additional functionality to set time contingents for customers.

60.1.1 Users, Groups & Roles

It is possible to set working hours per day to set a different value for the calculation of the new contingent.

Customers

After installation of the package a new field is added to the *Add Customer* and *Edit Customer* screens.



The screenshot shows the 'Add Customer' form with the following fields and controls:

- ★ CustomerID:
- ★ Customer:
- Street:
- Zip:
- City:
- Country:
- URL:
- Comment:
- ★ Valid:
- Workhours per day (e.g. format: 7.5):
- or

Fig. 1: Add Customer Screen

New Field

Workhours per day Add the working hours per day if it is different from the normal working day. If the field is left empty, 8 hours per day is calculated.

60.2 Agent Interface

This package extends the quota functionality to be able to set different quota for each customer.

60.2.1 Customers

This package extends the *Customer Information Center* screen with quota management.

Customer Information Center

After installation of the package two new widget will be displayed in the customer information center to set a quota for your customers. You can also set a reference number to recognize the purchase number of this contingent.

To set a quota for a customer:

1. Search for a customer in the *Customer Information Center*.
2. Add a reference number and new quota to the customer in the *Add quota to customer* widget in the right sidebar.

The screenshot shows a widget titled "Add quota to customer". It contains three input fields: "Reference#:" with an empty text box, "★ Time units:" with an empty text box, and "Time unit type:" with a dropdown menu currently showing "Hours". Below these fields is a "Create" button.

Fig. 2: Add Quota to Customer Widget

3. Check the entered data in the *Quota list* widget below.

The screenshot shows a widget titled "Quota list". It displays two lines of text: "PO 123: 03d 06h" and "Remaining: 03d 06h".

Fig. 3: Quota List Widget

You can set the quota in hours or days. The quota will be recalculated into days and hours in the *Quota list* widget.

A working day contains 8 working hours by default. The working hours per day can be changed in the *Customers* management screen.

The quota can be reduced by adding tickets or by adding articles to existing ticket in the *Ticket Zoom* screen.

60.2.2 Tickets

Use the *Ticket Zoom* screen to reduce the quota set for a customer.

Ticket Zoom

For each ticket you can set a time to reduce the quota of the customer. It is based on the work time units and there is a new drop-down field to reduce the contingent in hours or minutes.

To reduce quota in a ticket:

1. Create a ticket based for the customer.
2. Open the ticket in the *Ticket Zoom* screen. A new *Quota list* widget is also displayed in the right sidebar.
3. Add a note or any other communication article that has quota field.
4. Add a new time unit based on hours or minutes to reduce the quota.

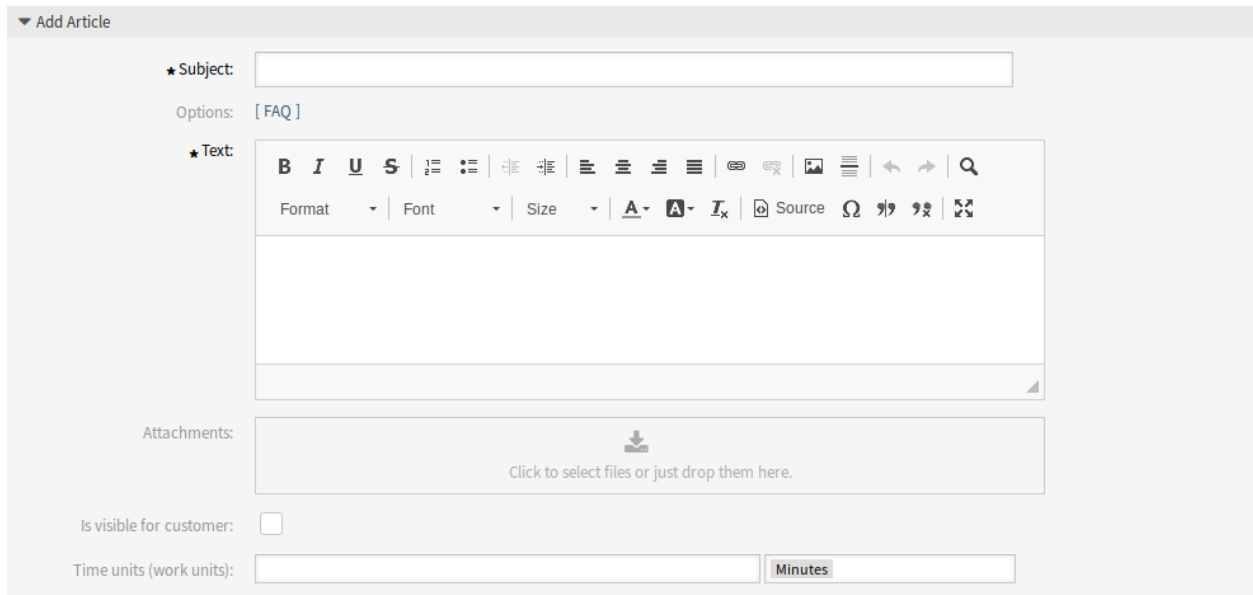


Fig. 4: Add Article Screen

Note: The quota can be also reduced in any ticket create screens.

60.3 External Interface

The quota of each customer is also displayed in the head of the external interface.

Service organizations who have to meet strict service level agreements (SLA) or those who want to react faster to requests from their most important customers should use this feature add-on.

Specific customer users can be defined as VIP customer users, which is shown in the customer information widget when a new ticket is created for him or the service staff is working on one of his existing tickets.

An additional list in system configuration allows an agent to assign the VIP status to several customer users.

Furthermore, SLAs can be restricted to be available to VIP customer users only. Keep an eye on your most important customers and deliver excellent service to them.

Benefits

- Respond faster to requests of specific customers.
- Facilitates SLA management.

Target Groups

- IT service management
- Call centers
- Service providers
- All companies working with SLAs

Available in Service Package

- SILVER

Package Name in OTRS Package Manager

- OTRSVIPCustomer

Note: Not compatible with the following feature add-on:

- *Restrict Customer Data View*
-

61.1 Administrator Interface

This package has no administrator interface.

61.1.1 System Configuration

The package has no default configuration for customer users and SLAs available only for VIP customer users. Customer users and SLAs have to be added in the system configuration.

To set VIP status for a customer user:

1. Go to *System Configuration* screen.
2. Select *OTRSVIPCustomer* in the *Navigation* widget.
3. Navigate to *Core* → *OTRSVIPCustomer* in the navigation tree.
4. Add the customer user IDs (login names) to setting `VIP : :UIDs`.

To set an existing SLA only available for VIP customer users:

1. Go to *System Configuration* screen.
2. Select *OTRSVIPCustomer* in the *Navigation* widget.
3. Navigate to *Core* → *OTRSVIPCustomer* in the navigation tree.
4. Add the SLAs to setting `VIP : :SLAs`.

Note: If a customer user is configured as a VIP, the internal customer record will have the attribute `UserVIP` set to `TRUE`. This allows to access the customer user's VIP status through other technical facilities, such as ACL definitions.

61.2 Agent Interface

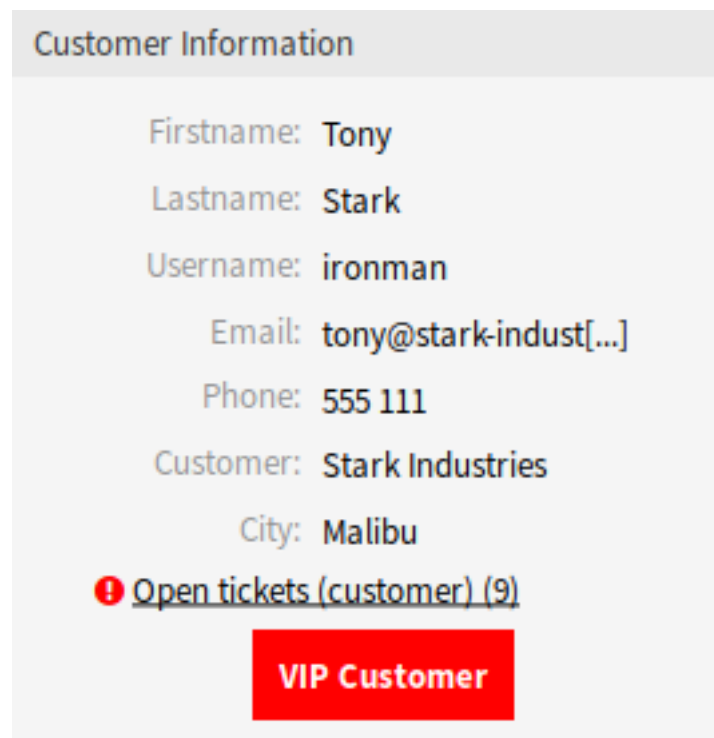
If a customer user is configured as a VIP, the *Customer Information* widget will show the VIP status of the user with a big red *VIP Customer* badge.

The extended *Customer Information* widget is available in all ticket create screens as well as in the *Ticket Zoom* screen.

If an SLA is configured to be only shown for VIP customer users, it will not be available anymore to regular users.

61.3 External Interface

This package has no external interface.



The image shows a 'Customer Information' widget with a light gray background and a darker gray header. The header contains the text 'Customer Information'. Below the header, the following information is displayed: 'Firstname: Tony', 'Lastname: Stark', 'Username: ironman', 'Email: tony@stark-indust[...]', 'Phone: 555 111', 'Customer: Stark Industries', and 'City: Malibu'. Below this information is a red circle with a white exclamation mark, followed by the text 'Open tickets (customer) (9)'. At the bottom of the widget is a red rectangular button with the white text 'VIP Customer'.

Customer Information

Firstname: Tony
Lastname: Stark
Username: ironman
Email: tony@stark-indust[...]
Phone: 555 111
Customer: Stark Industries
City: Malibu

! [Open tickets \(customer\) \(9\)](#)

VIP Customer

Fig. 1: Customer Information Widget