## CONTENTS

1 Introduction ............................................. 3  
   1.1 Categories .................................... 3  

2 Adjust Sort Ticket Overview ......................... 7  

3 Advanced Editor ...................................... 9  
   3.1 Administrator Interface .......................... 9  
   3.2 Agent Interface ................................ 11  
   3.3 External Interface ............................... 12  

4 Advanced Escalations ................................. 13  
   4.1 Administrator Interface ....................... 14  
   4.2 Agent Interface ................................ 34  
   4.3 External Interface ............................... 40  

5 Advanced Generic Agent .............................. 41  
   5.1 Administrator Interface ....................... 42  
   5.2 Agent Interface ................................ 42  
   5.3 External Interface ............................... 42  

6 Advanced Ticket Article Forward ................... 43  
   6.1 Administrator Interface ....................... 44  
   6.2 Agent Interface ................................ 44  
   6.3 External Interface ............................... 45  

7 Advanced Ticket Overview ............................ 47  
   7.1 Administrator Interface ....................... 48  
   7.2 Agent Interface ................................ 48  
   7.3 External Interface ............................... 48  

8 Agent Email Interface ................................. 49  
   8.1 Administrator Interface ....................... 50  
   8.2 Agent Interface ................................ 50  
   8.3 External Interface ............................... 51  

9 Automated FAQ Ticket Creator ...................... 53  
   9.1 Administrator Interface ....................... 54  
   9.2 Agent Interface ................................ 59  
   9.3 External Interface ............................... 61  

10 Automatic Start of Processes ....................... 63
| Section | 10.1 Administrator Interface | 10.2 Agent Interface | 10.3 External Interface | 11 Baramundi Inventory Connector | 11.1 Administrator Interface | 11.2 Agent Interface | 11.3 External Interface | 12 Calendar Resource Planning | 12.1 Administrator Interface | 12.2 Agent Interface | 12.3 External Interface | 13 Categories for Text Modules | 13.1 Administrator Interface | 13.2 Agent Interface | 13.3 External Interface | 14 CI Assignment Attribute Dynamic Field Map | 14.1 Administrator Interface | 14.2 Agent Interface | 14.3 External Interface | 15 CI References | 15.1 Administrator Interface | 15.2 Agent Interface | 15.3 External Interface | 16 CIs in Customer Frontend | 16.1 Administrator Interface | 16.2 Agent Interface | 16.3 External Interface | 17 Configuration Management Connector | 17.1 Administrator Interface | 17.2 Agent Interface | 17.3 External Interface | 18 Credit Card Filter | 18.1 Administrator Interface | 18.2 Agent Interface | 18.3 External Interface | 19 Custom Contact Fields | 19.1 Administrator Interface | 19.2 Agent Interface | 19.3 External Interface | 20 Customer Activity Widget | 20.1 Administrator Interface | 20.2 Agent Interface | 20.3 External Interface | 21 Customer Event Ticket Calendar | 21.1 Administrator Interface |
|---------|-----------------------------|---------------------|-----------------------|-------------------------------|-------------------------------|-----------------------------|-----------------------|-----------------------------|-------------------------------|-----------------------------|-----------------------------|-------------------------------|-----------------------------|-----------------------------|-----------------------------|-------------------------------|-----------------------------|-----------------------------|-----------------------------|-------------------------------|-----------------------------|-----------------------------|-----------------------------|-------------------------------|-----------------------------|-----------------------------|-----------------------------|-------------------------------|-----------------------------|-----------------------------|-----------------------------|-------------------------------|-----------------------------|-----------------------------|-----------------------------|
21.2 Agent Interface .......................................................... 153
21.3 External Interface ...................................................... 154

22 Customer Frontend Link Object 155
22.1 Administrator Interface ............................................ 156
22.2 Agent Interface .......................................................... 156
22.3 External Interface ...................................................... 156

23 Customer-specific Services 157
23.1 Administrator Interface ............................................ 158
23.2 Agent Interface .......................................................... 161
23.3 External Interface ...................................................... 161

24 Dashboard News Widget 163
24.1 Administrator Interface ............................................ 163
24.2 Agent Interface .......................................................... 167
24.3 External Interface ...................................................... 167

25 Data Privacy Protection 169
25.1 Administrator Interface ............................................ 170
25.2 Agent Interface .......................................................... 183
25.3 External Interface ...................................................... 183

26 Delete Attachments 185
26.1 Administrator Interface ............................................ 185
26.2 Agent Interface .......................................................... 188
26.3 External Interface ...................................................... 188

27 Dynamic Field Attachment 189
27.1 Administrator Interface ............................................ 189
27.2 Agent Interface .......................................................... 191
27.3 External Interface ...................................................... 193

28 Dynamic Field Calculation 195
28.1 Administrator Interface ............................................ 195
28.2 Agent Interface .......................................................... 197
28.3 External Interface ...................................................... 197

29 Dynamic Field CI 199
29.1 Administrator Interface ............................................ 200
29.2 Agent Interface .......................................................... 203
29.3 External Interface ...................................................... 204

30 Dynamic Field Database 205
30.1 Administrator Interface ............................................ 206
30.2 Agent Interface .......................................................... 210
30.3 External Interface ...................................................... 210

31 Dynamic Field Search Booster 211
31.1 Administrator Interface ............................................ 211
31.2 Agent Interface .......................................................... 212
31.3 External Interface ...................................................... 212

32 Dynamic Field Value Import 213
32.1 Administrator Interface ............................................ 214
32.2 Agent Interface .......................................................... 218
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>33 Dynamic Field Web Service</td>
<td>219</td>
</tr>
<tr>
<td>33.1 Administrator Interface</td>
<td>219</td>
</tr>
<tr>
<td>33.2 Agent Interface</td>
<td>228</td>
</tr>
<tr>
<td>33.3 External Interface</td>
<td>228</td>
</tr>
<tr>
<td>34 Dynamic Popup Window Size</td>
<td>229</td>
</tr>
<tr>
<td>35 Dynamic Sender Addresses</td>
<td>231</td>
</tr>
<tr>
<td>35.1 Administrator Interface</td>
<td>232</td>
</tr>
<tr>
<td>35.2 Agent Interface</td>
<td>235</td>
</tr>
<tr>
<td>35.3 External Interface</td>
<td>237</td>
</tr>
<tr>
<td>36 Escalation Suspend</td>
<td>239</td>
</tr>
<tr>
<td>36.1 Administrator Interface</td>
<td>240</td>
</tr>
<tr>
<td>36.2 Agent Interface</td>
<td>240</td>
</tr>
<tr>
<td>36.3 External Interface</td>
<td>242</td>
</tr>
<tr>
<td>37 Extended Ticket Stats</td>
<td>243</td>
</tr>
<tr>
<td>37.1 Administrator Interface</td>
<td>244</td>
</tr>
<tr>
<td>37.2 Agent Interface</td>
<td>244</td>
</tr>
<tr>
<td>37.3 External Interface</td>
<td>247</td>
</tr>
<tr>
<td>38 Focus Topics</td>
<td>249</td>
</tr>
<tr>
<td>38.1 Administrator Interface</td>
<td>249</td>
</tr>
<tr>
<td>38.2 Agent Interface</td>
<td>251</td>
</tr>
<tr>
<td>38.3 External Interface</td>
<td>253</td>
</tr>
<tr>
<td>39 Hide/Show Dynamic Fields</td>
<td>255</td>
</tr>
<tr>
<td>39.1 Administrator Interface</td>
<td>256</td>
</tr>
<tr>
<td>39.2 Agent Interface</td>
<td>262</td>
</tr>
<tr>
<td>39.3 External Interface</td>
<td>262</td>
</tr>
<tr>
<td>40 LDAP Password Notifications</td>
<td>263</td>
</tr>
<tr>
<td>40.1 Administrator Interface</td>
<td>264</td>
</tr>
<tr>
<td>40.2 Agent Interface</td>
<td>264</td>
</tr>
<tr>
<td>40.3 External Interface</td>
<td>264</td>
</tr>
<tr>
<td>41 Link Object Connector</td>
<td>265</td>
</tr>
<tr>
<td>41.1 Administrator Interface</td>
<td>265</td>
</tr>
<tr>
<td>41.2 Agent Interface</td>
<td>274</td>
</tr>
<tr>
<td>41.3 External Interface</td>
<td>275</td>
</tr>
<tr>
<td>42 MobyDick Connector</td>
<td>277</td>
</tr>
<tr>
<td>43 Out of Office</td>
<td>279</td>
</tr>
<tr>
<td>43.1 Administrator Interface</td>
<td>279</td>
</tr>
<tr>
<td>43.2 Agent Interface</td>
<td>280</td>
</tr>
<tr>
<td>43.3 External Interface</td>
<td>280</td>
</tr>
<tr>
<td>44 Primary / Secondary</td>
<td>281</td>
</tr>
<tr>
<td>44.1 Administrator Interface</td>
<td>281</td>
</tr>
<tr>
<td>44.2 Agent Interface</td>
<td>284</td>
</tr>
<tr>
<td>44.3 External Interface</td>
<td>287</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>57 Survey</td>
<td>335</td>
</tr>
<tr>
<td>57.1 Administrator Interface</td>
<td>335</td>
</tr>
<tr>
<td>57.2 Agent Interface</td>
<td>339</td>
</tr>
<tr>
<td>57.3 External Interface</td>
<td>339</td>
</tr>
<tr>
<td>58 System Configuration History</td>
<td>341</td>
</tr>
<tr>
<td>58.1 Administrator Interface</td>
<td>341</td>
</tr>
<tr>
<td>58.2 Agent Interface</td>
<td>345</td>
</tr>
<tr>
<td>58.3 External Interface</td>
<td>345</td>
</tr>
<tr>
<td>59 System Monitoring</td>
<td>347</td>
</tr>
<tr>
<td>59.1 Administrator Interface</td>
<td>347</td>
</tr>
<tr>
<td>59.2 Agent Interface</td>
<td>350</td>
</tr>
<tr>
<td>59.3 External Interface</td>
<td>350</td>
</tr>
<tr>
<td>60 Tag Cloud</td>
<td>351</td>
</tr>
<tr>
<td>61 Ticket Allocation</td>
<td>353</td>
</tr>
<tr>
<td>61.1 Administrator Interface</td>
<td>354</td>
</tr>
<tr>
<td>61.2 Agent Interface</td>
<td>360</td>
</tr>
<tr>
<td>61.3 External Interface</td>
<td>360</td>
</tr>
<tr>
<td>62 Ticket Forms</td>
<td>363</td>
</tr>
<tr>
<td>62.1 Administrator Interface</td>
<td>364</td>
</tr>
<tr>
<td>62.2 Agent Interface</td>
<td>377</td>
</tr>
<tr>
<td>62.3 External Interface</td>
<td>378</td>
</tr>
<tr>
<td>63 Ticket Invoker</td>
<td>379</td>
</tr>
<tr>
<td>63.1 Administrator Interface</td>
<td>379</td>
</tr>
<tr>
<td>63.2 Agent Interface</td>
<td>386</td>
</tr>
<tr>
<td>63.3 External Interface</td>
<td>387</td>
</tr>
<tr>
<td>64 Ticket Notification Popups</td>
<td>389</td>
</tr>
<tr>
<td>65 Ticket Queue Selection</td>
<td>391</td>
</tr>
<tr>
<td>65.1 Administrator Interface</td>
<td>392</td>
</tr>
<tr>
<td>65.2 Agent Interface</td>
<td>392</td>
</tr>
<tr>
<td>65.3 External Interface</td>
<td>392</td>
</tr>
<tr>
<td>66 Ticket Time Unit Dropdown</td>
<td>393</td>
</tr>
<tr>
<td>66.1 Administrator Interface</td>
<td>394</td>
</tr>
<tr>
<td>66.2 Agent Interface</td>
<td>394</td>
</tr>
<tr>
<td>66.3 External Interface</td>
<td>394</td>
</tr>
<tr>
<td>67 Ticket Watchlist</td>
<td>395</td>
</tr>
<tr>
<td>67.1 Administrator Interface</td>
<td>396</td>
</tr>
<tr>
<td>67.2 Agent Interface</td>
<td>397</td>
</tr>
<tr>
<td>67.3 External Interface</td>
<td>401</td>
</tr>
<tr>
<td>68 Ticket Workflow</td>
<td>403</td>
</tr>
<tr>
<td>69 Time Accounting</td>
<td>405</td>
</tr>
<tr>
<td>70 Time and Quota Management</td>
<td>407</td>
</tr>
<tr>
<td>70.1 Administrator Interface</td>
<td>408</td>
</tr>
<tr>
<td>70.2 Agent Interface</td>
<td>409</td>
</tr>
</tbody>
</table>
INTRODUCTION

This manual lists all feature add-ons.

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

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
- Automatic Start of Processes
- CI Assignment Attribute Dynamic Field Map
- Dynamic Sender Addresses
- Focus Topics
- LDAP Password Notifications
- Process Management Article Email
- Ready2Adopt ITSM Processes
- Ready2Adopt Processes
- Service-based Queue Routing
- Ticket Allocation
• Ticket Queue Selection

Customer Management
• Custom Contact Fields
• Customer Activity Widget
• Customer-specific Services
• Time and Quota Management

Individualization
• Advanced Ticket Overview
• Hide/Show Dynamic Fields
• Specific Ticket Notifications
• Ticket Forms

Integration
• Baramundi Inventory Connector
• Configuration Management Connector
• Dynamic Field Database
• Dynamic Field Web Service
• Link Object Connector
• Ready2Adopt Web Services
• SAP Solution Manager Connector
• System Monitoring
• Ticket Invoker

Knowledge Management & Self Service
• CI References
• CIs in Customer Frontend
• Customer Event Ticket Calendar
• Customer Frontend Link Object
• Dashboard News Widget

Performance
• Dynamic Field Search Booster

Reporting
• Extended Ticket Stats
• Survey

Security & Permissions
• Credit Card Filter
• Data Privacy Protection
• Delete Attachments
• Restrict Customer Data View

Ticket Management
• Advanced Ticket Article Forward
• Categories for Text Modules
• Dynamic Field Attachment
• Dynamic Field Calculation
• Dynamic Field CI
• Out of Office
• Primary / Secondary
• Restore Pending Information
• Service Categories
• State Pre-selection Response Templates
• Ticket Watchlist

Time Management
• Calendar Resource Planning
• Advanced Escalations
• Escalation Suspend
• Ticket Time Unit Dropdown
CHAPTER TWO

ADJUST SORT TICKET OVERVIEW

**Warning:** This feature add-on has reached the end of life with OTRS 7 and will not be longer supported.

The functionality for adjusting the column width has been integrated into the OTRS framework. The width of the columns in the ticket lists is now individually configurable and can even vary between the different static and personal lists. The word length of the title column in the ticket lists now automatically adjusts to the column size.

The abbreviated display of queue and service names has been removed. Instead, only the names without the full path are displayed. You can get more details about the full path by hovering over the name in the lists. The requirement to display the ticket number has been removed.
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.
- 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

- SILVER, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager

- OTRSAdvancedEditor

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).
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.

```plaintext
[% IF ( matches = Data.CustomerUser.UserLanguage.match('^de') ) %]
[% ELSIF ( matches = Data.CustomerUser.UserLanguage.match('^hu') ) %]
[% ELSE %]
[% END %]
```

7. Click on the **OK** button to save the code snippet.

```plaintext
[% IF ( matches = Data.CustomerUser.UserLanguage.match('^de') ) %]
[% ELSIF ( matches = Data.CustomerUser.UserLanguage.match('^hu') ) %]
[% ELSE %]
[% END %]
```

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.

### 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.

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

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 previous page.
2. Edit the template and save it.
3. Repeat these steps until no more errors are found.

### 3.2 Agent Interface

This package has no agent interface.
3.3 External Interface

This package has no external interface.
This feature add-on makes your escalation management more flexible and adjusts it according to your customers or to different service level agreements. Escalation types defined in OTRS, such as First Response Time, Update Time, and Solution Time, can be enhanced by creating new types and defining your own names and properties.

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

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

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

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

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

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

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

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

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

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

Available in Service Package
• PLATINUM

Package Name in OTRS Package Manager
• OTRSAdvancedEscalations

Note: Not compatible with the following feature add-ons:
• Advanced Ticket Overview
• 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 have to be connected to SLAs, and it is possible to filter them by 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.

![Escalation Type Bundle Management Screen](image)

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:

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**

Longer text can be added here to describe the escalation type bundle.

**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** *

The execution order takes place, when at least two bundles could start based on the types and the attributes of the ticket. Only one bundle can be used though. The execution order will change the priority of the bundle, so that the bundle with higher priority will be used.
4.1. Administrator Interface

Fig. 3: Edit Escalation Type Bundle Screen

Fig. 4: Delete Escalation Type Bundle Screen
Validity *
Set the validity of this resource. Each resource can be used in OTRS only, if this field is set to valid. Setting this field to invalid or invalid-temporarily will disable the use of the resource.

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. If 1 day is set, the escalation will start on the next working day (based on the working time).

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 and the time when the color of the escalation will change to orange.

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.

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.

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.
Fig. 7: Edit Escalation Type Screen
Escalation Type Settings

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

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

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

Description
Like comment, but longer text can be added here.

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).

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 Screen](image)

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 and the times 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).

Note: It is not needed to adjust the CustomEscalation in the notification tag, but it always takes the information of the used escalation event. If First Response (NotifyBefore) is used it will show the information of the first response.
Escalation Events

Escalation events of the advanced escalation always start with \textit{Escalation: Escalation Type Name (Event)}. The following escalation events are available:

\textbf{Escalation: Escalation Type Name (Start) (EscalationStart\_[EscalationTypeID])}
This event will be executed if the escalation has been started.

\textbf{Escalation: Escalation Type Name (NotifyBefore) (EscalationNotifyBefore\_[EscalationTypeID])}
This event will be executed if the time reached where the agent gets notified for the escalation.

\textbf{Escalation: Escalation Type Name (Breached) (EscalationBreached\_[EscalationTypeID])}
This event will be executed if the escalation time has been reached.

\textbf{Escalation: Escalation Type Name (Restart) (EscalationRestart\_[EscalationTypeID])}
This event will be executed if the escalation has been restarted.

\textbf{Escalation: Escalation Type Name (Suspend) (EscalationSuspend\_[EscalationTypeID])}
This event will be executed if the escalation has been suspended.

\textbf{Escalation: Escalation Type Name (ResumeSuspend) (EscalationResumeSuspend\_[EscalationTypeID])}
This event will be executed if the escalation has been resumed from a suspend state.

\textbf{Escalation: Escalation Type Name (ResumeStop) (EscalationResumeStop\_[EscalationTypeID])}
This event will be executed if the escalation has been resumed from a stop state.

\textbf{Escalation: Escalation Type Name (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.

\textbf{escalation\_history Table}

All escalation events will create new entries in the \textit{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:

\begin{itemize}
  \item \texttt{Ticket::EventModulePost###EscalationHistory (group: OTRSAdvancedEscalations, navigation: Core → Event → EscalationHistory)}.
\end{itemize}

Make sure OTRS daemon is running.

\begin{verbatim}
shell> /opt/otrs/bin/otrs.Daemon.pl status
\end{verbatim}

This event module will track the following escalation events:

\begin{itemize}
  \item EscalationStart
\end{itemize}
- 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.
<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>notify_remaining_wt</td>
<td>This column contains the seconds until the notify start after a suspend of an escalation type (calculated with working calendars).</td>
</tr>
<tr>
<td>running_total_time</td>
<td>This column contains the total amount of seconds the timer was running based on the Timer(Start</td>
</tr>
<tr>
<td>running_total_wt</td>
<td>This column contains the total amount of seconds the timer was running based on the Timer(Start</td>
</tr>
<tr>
<td>running_total_virtual_time</td>
<td>This column contains the total amount of seconds the timer was running based on history entries.</td>
</tr>
<tr>
<td>running_total_virtual_wt</td>
<td>This column contains the total amount of seconds the timer was running based on history entries (calculated with working calendars).</td>
</tr>
<tr>
<td>suspend_total_time</td>
<td>This column contains the total amount of suspended seconds of the escalation type based on the Timer(Start</td>
</tr>
<tr>
<td>suspend_total_wt</td>
<td>This column contains the total amount of suspended seconds of the escalation type based on the Timer(Start</td>
</tr>
<tr>
<td>running_last_time</td>
<td>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).</td>
</tr>
<tr>
<td>running_last_wt</td>
<td>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).</td>
</tr>
<tr>
<td>running_last_virtual_time</td>
<td>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.</td>
</tr>
<tr>
<td>running_last_virtual_wt</td>
<td>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).</td>
</tr>
<tr>
<td>suspend_last_time</td>
<td>This column contains the amount of seconds the ticket escalation was suspended last time based on the history entries of the ticket.</td>
</tr>
<tr>
<td>suspend_last_wt</td>
<td>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).</td>
</tr>
</tbody>
</table>
create_time
This column contains create time of the escalation history entry.

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

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

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

escalation_history_data Table
All escalation events will create new entries in the escalation_history table. For each escalation event it is possible to save ticket and dynamic field data in a separate data table. Make sure that TriggerEscalationStartEvents is enabled. The attributes which will be saved can be configured in the following system configuration options:

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

  Example configuration: Queue → 1

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

  Example configuration: DynamicField_Test → 1

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

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

Make sure OTRS daemon is running.

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

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

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

escalation_history_id
This column contains ID of the related escalation_history entry.

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

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

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

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

change_time
This column contains date and time when the escalation history data set was changed.
**change_by**
This column contains ID of the user who triggered the data set change.

**System Configuration**

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

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

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

In addition to this, you have the possibility to specify which advanced escalation columns you want to use in the ticket lists. These possible advanced escalation columns are not active by default. If you want to use them, you have to switch them on explicitly. Please follow instructions as outlined in the sections below.

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

**Display Advanced Escalation Columns**

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

**Dashboard Ticket List Widgets**

In the following example, we will add an advanced escalation column named _EscalationDatetime_ to the _Escalations_ widget on the dashboard, for all available escalation types.

1. Go to the _System Configuration_ screen.
2. Search for the setting _AgentFrontend::Dashboard::Widget###EscalatedTickets_.
3. Under the _Config_ key, add the following keys to the existing YAML configuration:

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

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

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

---

4.1. Administrator Interface 29
5. Click on the tick button on the right to save the setting.
6. Deploy the modified system configuration.

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

---

### Ticket List Screens

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

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

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

4. Make sure to just add the new column name to the existing structure, taking care to follow the rules of the YAML syntax.
5. Click on the tick button on the right to save the setting.
6. Deploy the modified system configuration.

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

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

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

```
EscalationType_ColumnIdentifier
```

Where:

**ColumnIdentifier**

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

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

```
EscalationType_EscalationRemainingTime
```

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

```
Escalation_ID_ColumnIdentifier
```

Where:

**ID**

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

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

In the example above, the ID is 1.

**ColumnIdentifier**

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

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

```
Escalation_1_EscalationWorkingTime
```

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

```
Escalation_ID:
  IsVisible: 0|1|2
```

Where:

**ID**

Specific escalation type ID.

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

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

In the example above, the ID is 1.

**IsVisible**

Defines whether the column is not visible (0), not visible by default but the agent can make it visible (1) or visible by default (2).

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

```
Escalation_1:
  IsVisible: 2
```

**Possible Advanced Escalation Columns**

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

**EscalationDatetime**

Timestamp of the escalation date.

**EscalationReached**

Yes/no value whether escalation time is reached.

**EscalationTime**

Seconds until the escalation type is breached.

**EscalationWorkingTime**

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

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

**NotifyDateTime**
Timestamp of notify start (only set if given).

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

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

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

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

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

**SuspendTotalTime**
Total amount of seconds that the timer was suspended.

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

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

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

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

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

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

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

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

**RunningLastVirtualWorkingTime**
Total amount of seconds that the timer was running last time with BaseTime as start date (excluding suspend times) with working calendars (only if calendar is defined).
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 lists, ticket list widgets and statistics. The advanced escalations data is calculated on-the-fly and could have a huge negative performance impact on the whole system.

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

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

Display Escalations in a Widget

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

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

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

Debug Escalations

Advanced escalations might be very complex and it needs to be investigated if something does not work as expected. 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 5 seconds of the ticket history by default, but it can be adjusted in the system configuration.

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

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

It is usually in Escalation Type Name – Escalation Bundle Name format. In the example above the escalation type name is First Response and the escalation bundle name is Bundle.

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

**Select Escalation Types**

- First Response - Bundle
- Update Time - Bundle
- Solution Time - Bundle

New Ticket (2015071510123456) created.

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

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

**Debug - Start - First Response - Bundle**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Currently</th>
<th>Previous</th>
<th>Was</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority</td>
<td>3 normal</td>
<td>3 normal</td>
<td></td>
</tr>
<tr>
<td>Queue</td>
<td>Raw</td>
<td>Raw</td>
<td></td>
</tr>
<tr>
<td>StateType</td>
<td>new</td>
<td>new</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>new</td>
<td>new</td>
<td></td>
</tr>
<tr>
<td>TicketType</td>
<td>Unclassified</td>
<td>Unclassified</td>
<td></td>
</tr>
</tbody>
</table>

**Attributes**

- Agent Message Present, Js_StateType, WasNever_StateType

**Fixed Filters**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Filters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Agent Message Present</td>
</tr>
<tr>
<td></td>
<td>Agent Message Sent</td>
</tr>
<tr>
<td></td>
<td>Created By Customer</td>
</tr>
<tr>
<td></td>
<td>Customer Message Present</td>
</tr>
<tr>
<td></td>
<td>Customer Message Sent</td>
</tr>
</tbody>
</table>

---

**Fig. 15: Escalation Debug Mode – Select Escalation Type**

**Fig. 16: 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 successful (success = green, red = no change of the state). Basically, for each snapshot it is possible that the escalation will calculate multiple state if it does not match to the first.

My %EscalationStatesMachine = {
   NotRunning => [ 'Start', 'Restart' ],
   Running    => [ 'Suspend', 'Restart', 'Stop' ],
   Suspended  => [ 'Resume', 'Stop' ],
};

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

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 4 columns:

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

---

4.2. Agent Interface
Is \rightarrow "equals now",
IsNot \rightarrow "doesn't equal now",
Was \rightarrow "equaled",
WasNever \rightarrow "did never equal",
ChangedTo \rightarrow "changed to",
IsPrev \rightarrow "equals previous",
IsNotPrev \rightarrow "doesn't equal previous",

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

**Escalation Attributes**

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

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Currently</th>
<th>Previous</th>
<th>Was</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority</td>
<td>3 normal</td>
<td>3 normal</td>
<td></td>
</tr>
<tr>
<td>Queue</td>
<td>Raw</td>
<td>Raw</td>
<td></td>
</tr>
<tr>
<td>StateType</td>
<td>open</td>
<td>new</td>
<td>new</td>
</tr>
<tr>
<td>State</td>
<td>open</td>
<td>new</td>
<td>new</td>
</tr>
<tr>
<td>TicketType</td>
<td>Unclassified</td>
<td>Unclassified</td>
<td></td>
</tr>
</tbody>
</table>

**Fig. 19: Escalation Debug Mode – Escalation Attributes**

There are 4 columns:

- **Attribute**: This column contains the ticket attribute.
- **Currently**: This column contains the current value of the snapshot. This value will be used for the calculation of \textit{equals now}, \textit{doesn't equal now} and \textit{changed to} conditions.
- **Previous**: This column contains the value of the previous snapshot. This value will be used for the calculation of \textit{equals previous}, \textit{doesn't equal previous} and \textit{changed to} conditions.
- **Was**: This column contains the value of all previous snapshot. This value will be used for the calculation of \textit{did never equal} and \textit{equaled} conditions.

**Escalation Filters**

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

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

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

Advanced Escalations Data in Statistics

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

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

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

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

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

Filtering

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

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

**Datetime columns**
For this type of columns (e.g. EscalationDatet ime), 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.
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**

- SILVER, GOLD, TITANIUM, PLATINUM

**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](image)

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.
ADVANCED TICKET ARTICLE FORWARD

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

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

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

Benefits

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

Target Groups

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

Available in Service Package

- SILVER, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager

- OTRSAdvancedTicketArticleForward
6.1 Administrator Interface

This package has no administrator interface.

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

6.2 Agent Interface

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

6.2.1 Ticket Detail View

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

![Communication Stream Widget](image)

Fig. 1: Advanced Article Forward in Communication Stream Widget

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

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

Use case example:

1. The agent opens the ticket detail view to forward multiple articles. With the checkboxes in the article list, the relevant articles can be selected or all articles with the checkbox in the header row. The advanced article forward icon is disabled, if no article in the article list is selected. The agent can also forward a note if this possibility is activated in the system configuration settings.

2. The advanced article forward icon will be activated, if the agent selects at least one article.

3. In the Forward via Email action the selected articles are added as attachments.

4. If the feature is activated in the system configuration setting, the agent can set all forwarded articles as seen with sending the forward mail.
6.3 External Interface

This package has no external interface.
ADVANCED TICKET OVERVIEW

With this feature add-on you receive an additional way to view linked items. To see linked objects and tickets, you just have to move the mouse over the chain icon in the Related column of your ticket list, which is visible if the ticket contains linked items of any sort. Hovering the mouse cursor over the chain icon will open a pop-up showing all linked items grouped by type. To name a few: appointments, configuration items, knowledge items, or linked tickets.

In the pop-up you can directly open the drop-down of each linked item group by clicking on them. You can set the columns for each object and of course, it is possible to directly open the linked ticket or object from the pop-up. So with the new extended ticket overview, agents get a much easier overview of linked items, saving them a lot of time and effort.

Benefits

- Time saving.
- Less effort.
- Better overview.

Target Groups

- Customer service
- IT service
- Sales departments
- Facility management
- Human resources
- Logistics

Available in Service Package

- SILVER, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager

- OTRSAdvancedTicketOverview
7.1 Administrator Interface

This package has no administrator interface.

7.2 Agent Interface

This feature adds a popover view for linked items inside ticket lists.

![Fig. 1: Related Items Inside Ticket List](image)

To see the related items, hover the mouse over the chain symbol in the column *Related*. The chain symbol is displayed only if the ticket has linked items. A popover window opens and displays the linked items grouped by types.

**Note:** This feature does not work on mobile devices.

7.3 External Interface

This package has no external interface.
CHAPTER
EIGHT

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:

\[<\text{OTRS\_CMD}>\text{send} </\text{OTRS\_CMD}>\]

or

\[<\text{OTRS\_CMD}>\text{send, nocc} </\text{OTRS\_CMD}>\]

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

The following commands exist:

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

\textbf{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.

\textbf{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.

\textbf{Target Groups}

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

\textbf{Available in Service Package}

- SILVER, GOLD, TITANIUM, PLATINUM

\textbf{Package Name in OTRS Package Manager}
• OTRSAgentEmailInterface

8.1 Administrator Interface

This package has no administrator interface.

8.1.1 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 the 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.

8.2 Agent Interface

This package has no agent interface.

8.2.1 Usage

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

```xml
<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.
8.3 External Interface

This package has no external interface.
AUTOMATED FAQ TICKET CREATOR

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

**Example**

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

**Benefits**

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

**Target Groups**

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

**Available in Service Package**

- SILVER, GOLD, TITANIUM, PLATINUM

**Package Name in OTRS Package Manager**

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

**OTRSAutomatedFAQTicketCreatorStartTime**

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

**OTRSAutomatedFAQTicketCreatorEndTime**

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

**OTRSAutomatedFAQTicketCreatorFrequency**

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

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

**OTRSAutomatedFAQTicketCreatorRepeatTimes**

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

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

**OTRSAutomatedFAQTicketCreatorRepeatOnDays**

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

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

**OTRSAutomatedFAQTicketCreatorLastExecution**

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

**OTRSAutomatedFAQTicketCreatorSubject**

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

See also:

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

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

Fig. 1: Dynamic Field Management Screen

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

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

• OTRSAutomatedFAQTicketCreator::DefaultValues###Title

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

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

• OTRSAutomatedFAQTicketCreator::DefaultValues###CustomerUser

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

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

• OTRSAutomatedFAQTicketCreator::DefaultValues###Owner

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

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

• OTRSAutomatedFAQTicketCreator::DefaultValues###Responsible

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

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

• OTRSAutomatedFAQTicketCreator::DefaultValues###Priority

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

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

• OTRSAutomatedFAQTicketCreator::DefaultValues###State

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

See also:
To define a default value for this dynamic field, you can also set a value to the following system configuration option:
• OTRSAutomatedFAQTicketCreator::DefaultValues###Service

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

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

• OTRSAutomatedFAQTicketCreator::DefaultValues###Body

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

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

• OTRSAutomatedFAQTicketCreator::DefaultValues###SLA

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

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

• OTRSAutomatedFAQTicketCreator::DefaultValues###Type

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

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

• OTRSAutomatedFAQTicketCreator::DefaultValues###Queue

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

**System Configuration**

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

• OTRSAutomatedFAQTicketCreator::Core::DynamicFieldMapping###DynamicField

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

• OTRSAutomatedFAQTicketCreator::DefaultValues###DynamicField
Usage

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

To edit field values of a dynamic field:

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

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

![Dropdown Field Settings](image)

Fig. 2: Edit Dynamic Field Value

To edit default field values of a dynamic field:

1. Go to the System Configuration screen.
3. Navigate to Core → OTRS Automated FAQ Ticket Creator → DefaultValues in the navigation tree.
4. Add the default value for customer user to setting OTRS Automated FAQ Ticket Creator::DefaultValues###CustomerUser.

![System Configuration – Default Value](image)

Fig. 3: System Configuration – Default Value

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

1. Go to the System Configuration screen.
3. Navigate to Core → OTRS Automated FAQ Ticket Creator in the navigation tree.
4. Search for setting OTRS Automated FAQ Ticket Creator::Core::DynamicFieldMapping###DynamicField.
5. Set mapping for a dynamic field. You need to use the ticket field as key and the knowledge base article field as value.
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 the 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`

### 9.2 Agent Interface

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

#### 9.2.1 Knowledge Base Articles

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

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

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

To show the dynamic fields that set ticket values:

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

Use Case Example

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

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

1. Create a new knowledge base article.
2. Set 2020-09-15 10:00:00 for Ticket create start time.
3. Set 2021-09-15 10:00:00 for Ticket create end time.
4. Set Daily for Ticket create frequency.
5. Set 01 for Ticket create repeat times.
6. Select Monday and Friday for Ticket create repeat on days.
7. Fill the following fields to define the values for the ticket which will be created:
   - Ticket create customer user
   - Ticket create owner
   - Ticket create responsible
   - Ticket create ticket type
   - Ticket create queue
• Ticket create priority
• Ticket create service
• Ticket create SLA
• Ticket create state

![Automated Knowledge Base Article Ticket Creator Settings](image)

Fig. 6: Create Knowledge Base Article

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

**Warning:** Times scheduled during daylight saving time start and daylight saving time end can have unexpected effects. At daylight saving time start it can be skipped and at daylight saving time end it can be executed twice. It is highly not recommended to schedule times when the daylight saving time starts or ends according to the server timezone.

### 9.3 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

- SILVER, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager

- OTRSAutomaticStartOfProcesses

10.1 Administrator Interface

This package implements a scheduled automatic creation of process tickets.
10.1.1 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.

![Available Process Elements]

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.

To edit an already defined start event, just click on its name in the *Available Process Elements* widget.
Fig. 2: Create New Start Event Screen

Fig. 3: Edit Start Event Screen
**Warning:** Times scheduled during daylight saving time start and daylight saving time end can have unexpected effects. At daylight saving time start it can be skipped and at daylight saving time end it can be executed twice. It is highly not recommended to schedule times when the daylight saving time starts or ends according to the server timezone.

**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.

![Canvas Diagram](image)

**Fig. 4: Start Event on Canvas**

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.
10.2 Agent Interface

This package has no agent interface.

10.3 External Interface

This package has no external interface.
This package contains functionality to synchronize configuration items from a baramundi Inventory instance with the OTRS CMDB.

**Available in Service Package**
- SILVER, GOLD, TITANIUM, PLATINUM

**Package Name in OTRS Package Manager**
- OTRSBaramundiInventoryConnector

---

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

---

### 11.1 Administrator Interface

This package enables OTRS to perform requests toward a remote *baramundi Inventory* installation. Only unidirectional communication with the controllers *Endpoints, SoftwareScanRules* and *EndpointInvSoftware* is supported. This allows to synchronize computers, software and their links into the OTRS CMDB.

The package provides a console command to manually or automatically trigger the supported controllers in order to keep the systems synchronized.

#### 11.1.1 Processes & Automation

This package adds a new web service and a network transport to the system.

**Web Services**

A new web service has been created automatically on package installation.
Configuration

To configure the baramundi Inventory remote web service:

1. Go to the Web Service Management screen and look for a web service named Baramundi.
2. Click on the Baramundi web service to open for configuration.
3. Click on the Configure button to edit the network transport in the section OTRS as requester next to the selected network transport HTTP::RESTBarmundi.
4. Change the endpoint to match your baramundi Inventory server and set correct values for BasicAuth user and password.
5. Click on the Save or Save and finish button.

If you are satisfied with the functionality at a later time it is recommended to change the debug threshold on the web service to Error. This considerably reduces logging and potentially improves the web service performance.

Triggering the Remote Web Service Manually

The package provides a console command to trigger the three supported controllers by specifying the requested controller as an argument.

**Note:** This feature is only available to On-Premise customers. If you are a Managed customer, this feature is taken care of by the Customer Solutions Team in OTRS. Please contact us via support@otrs.com or in the OTRS Portal.

The controller Endpoints retrieves devices/clients managed by baramundi and synchronizes them into the CMDB class BaramundiCI.

```
bin/otrs.Console.pl Maint::BaramundiInventory::Trigger Endpoints
```

The controller SoftwareScanRules retrieves software managed by baramundi and synchronizes them into the CMDB class BaramundiCISoftware.

```
bin/otrs.Console.pl Maint::BaramundiInventory::Trigger SoftwareScanRules
```

The controller EndpointInvSoftware retrieves associations between devices and software managed by baramundi and synchronizes them into OTRS via DependsOn/RequiredFor links between the configuration items.

```
bin/otrs.Console.pl Maint::BaramundiInventory::Trigger EndpointInvSoftware
```

Please note that execution is asynchronous and requests might take several minutes to be completed (especially the EndpointInvSoftware controller), depending on the amount of data provided by baramundi Inventory.

Correct execution can be checked via the generic interface debugger. By default response data exceeding 200 kB is suppressed. If you desire to raise or lower this limit, please change the configuration value for GenericInterface::Operation::ResponseLoggingMaxSize in the system configuration.

OTRS prevents simultaneous execution of more than one controller of each type automatically.
Fig. 1: Baramundi Web Service
### Network Transport Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>HTTP:REST</td>
</tr>
<tr>
<td>Endpoint</td>
<td><code>https://192.168.0.1:10070/bConnect/v1.0</code></td>
</tr>
<tr>
<td>Timeout</td>
<td>300</td>
</tr>
<tr>
<td>Authentication</td>
<td>BasicAuth</td>
</tr>
<tr>
<td>BasicAuth User</td>
<td><code>user@HOSTNAME</code></td>
</tr>
<tr>
<td>BasicAuth Password</td>
<td><code>**********</code></td>
</tr>
<tr>
<td>Use Proxy Options</td>
<td>No</td>
</tr>
<tr>
<td>Use SSL Options</td>
<td>No</td>
</tr>
<tr>
<td>Controller mapping for Invoker</td>
<td><code>/EndpointInvSoftware</code></td>
</tr>
</tbody>
</table>

**Baramundi EndpointInvSoftware**: The controller that the invoker should send requests to. Variables marked by a `:` will get replaced by the data value and passed along with the request. (e.g. `/Ticket/TicketId?UserLogin=UserName&Password=Password`).

**Valid request command for Invoker**: 

**Baramundi Endpoints**: A specific HTTP command to use for the requests with this Invoker (optional).

**Controller mapping for Invoker**: `/Endpoints`

**Baramundi Endpoints**: The controller that the invoker should send requests to. Variables marked by a `:` will get replaced by the data value and passed along with the request. (e.g. `/Ticket/TicketId?UserLogin=UserName&Password=Password`).

**Valid request command for Invoker**: 

**Baramundi Software Scan Rules**: A specific HTTP command to use for the requests with this Invoker (optional).

**Controller mapping for Invoker**: `/SoftwareScanRules`

**Baramundi Software Scan Rules**: The controller that the invoker should send requests to. Variables marked by a `:` will get replaced by the data value and passed along with the request. (e.g. `/Ticket/TicketId?UserLogin=UserName&Password=Password`).

**Valid request command for Invoker**: 

**Default command**: `GET`

The default HTTP command to use for the requests.
Triggering the Remote Web Service Automatically

If you would like the controllers to be run periodically in order to keep OTRS synchronized with baramundi Inventory, the daemon tasks provided by this package can be enabled in the Daemon::SchedulerCronTaskManager::Task###Baramundi system configuration. Please adjust the default frequency of 20 minutes to your requirements.

**Warning:** High frequencies might affect the system performance.

11.1.2 CMDB Settings

This package adds two new configuration item classes to the CMDB.

Configuration Items

Use this screen to manage class definition of configuration item classes. The configuration item class management screen is available in the Configuration Items module of the CMDB Settings group.

![Configuration Item Management Screen](image)

**New Classes**

**BaramundiCI**
Class for the baramundi Inventory computers.

**BaramundiCISoftware**
Class for the baramundi Inventory software.
11.2 Agent Interface

This package has no agent interface.

11.3 External Interface

This package has no external interface.
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, GOLD, TITANIUM, PLATINUM

**Package Name in OTRS Package Manager**

- OTRSCalendarResourcePlanning
12.1 Administrator Interface

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

12.1.1 Administration

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

Calendars

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

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

To set default values for appointments:

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

Manage Settings

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

Dynamic Fields

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

Each appointment dynamic fields are listed in this section. This feature controls only the visibility of the dynamic fields.

Possible options:

Hide
The dynamic field is hidden in the forms and in the appointments.

Show
The dynamic field is shown in the forms and in the appointments, but it can have an empty value.

Show as mandatory
The dynamic field is shown in the forms and in the appointments, and it does not allow an empty value.

Use form configuration
The display of the dynamic field will follow the current form configuration, that is defined in the following system configuration settings:
Fig. 1: Manage Settings Widget
In case the Hide, Show or Show as mandatory options are used, form configuration will be effectively ignored, and the dynamic field configuration from the calendar will be applied.

**Advanced Settings**

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

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

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

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

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

---

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

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

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

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

**Team Management**

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

To create a new team:

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

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

To edit a team:
1. Click on a team in the list of teams.
2. Modify the fields.
3. Click on the Save or Save and finish button.

![Edit Team Screen](image)

Fig. 4: Edit Team Screen

To manage team agents:

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

To export a team:

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

![List of Teams](image)

Fig. 5: Export Team Screen

To import a team:

1. Click on the Browse... button of the Team Import widget in the left sidebar.
2. Select a previously exported .yml file.
3. Click on the Overwrite existing entities checkbox, if you would like to overwrite the existing teams.
4. Click on the Import team button.

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

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

**Name** *

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

**Permission group** *

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

**Comment**

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

**Validity** *

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

Team List Management

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

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.

Fig. 7: Manage Team Agents Screen

Manage Agents ↔ Teams Relations

To assign some agents to a team:

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

Fig. 8: Change Team Relations for Agent

Fig. 9: Change Agent Relations for Team
Note: If several agents or teams are added to the system, use the filter box to find a particular agent or team by just typing the name to filter.

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

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

### 12.1.2 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.

![Appointment Filter Section](image)

**Fig. 10:** Appointment Filter Section

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

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

To activate the sending of ICS files:

1. Create a new appointment notification or select an existing appointment notification.
2. In the Events widget select AppointmentCreate, AppointmentUpdate or AppointmentDelete as event.
3. In the Notification Methods widget select Send ICS file.

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

12.1.3 Processes & Automation

This package adds support for dynamic fields to appointments and adds three modules to script task activities and sequence flow actions of process management.

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, Show as mandatory or Use form configuration.

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

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

Process Management

After installation of the package three new modules are added to script task activities and sequence flow actions of process management.
Process Modules

To see the new modules:

2. Create a new process or select an existing process that contains a script task activity.
3. Click on the Activities item in the Available Process Elements widget in the left sidebar.
4. Create a new script task activity or edit an existing one.
5. Select one of the new scripts in the Script drop-down.
   - AppointmentDataPull
   - AppointmentDataPush
   - LinkWithAppointment
6. Click on the Save button, if the Configure button is not visible next to the Script drop-down.
7. Click on the Configure button to add parameters (key-value pairs) for the script.

AppointmentDataPull

A module to fetch data from a linked appointment.

Fig. 11: Process Management Module AppointmentDataPull

Linked appointment condition section

This section is used to search for appointments.

Filters can be added with key-value pairs. There is logical AND relation between the filters if more than one filter is added. Logical OR relation can be added by multiple values separated by ,.

Desired behavior section

If more than one appointments are found in the section above, the desired behavior can be defined here.

Possible values:

- Copy attributes from the appointment that was found first
- Copy attributes from the appointment that was found last
- Ignore appointment, do not copy anything
Process ticket attributes section

With this module the process ticket attributes can be updated. The key is the attribute of the process ticket. The value can be a pre-defined text, an attribute from the linked appointment in form of an OTRS tag or a concatenation of both. The `<OTRS_APPOINTMENT_*>` OTRS tag prefix can be used here.

Examples:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority</td>
<td>5 very high</td>
</tr>
<tr>
<td>DynamicField_Location</td>
<td><code>&lt;OTRS_APPOINTMENT_Location&gt;</code></td>
</tr>
<tr>
<td>Title</td>
<td>From: <code>&lt;OTRS_APPOINTMENT_Title&gt;</code></td>
</tr>
</tbody>
</table>

See also:

See the `AppointmentDataPull` and the `AppointmentSearch()` API reference.

AppointmentDataPush

A module to insert data to a linked appointment.

Fig. 12: Process Management Module AppointmentDataPush

Linked appointment condition section

This section is used to search in the linked appointments.

Filters can be added with key-value pairs. There is logical AND relation between the filters if more than one filter is added. Logical OR relation can be added by multiple values separated by , .

Linked appointment attributes section

Here can be set the linked appointment attributes to be updated. The key is the attribute of the linked appointment. The value can be a pre-defined text, an attribute from the process ticket in form of an OTRS tag or a concatenation of both. The `<OTRS_TICKET_*>` OTRS tag prefix can be used here.

Examples:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>state</td>
<td>closed successful</td>
</tr>
<tr>
<td>Priority</td>
<td><code>&lt;OTRS_TICKET_Priority&gt;</code></td>
</tr>
</tbody>
</table>

See also:

See the `AppointmentDataPush` and the `AppointmentSearch()` API reference.
**LinkWithAppointment**

A module to link appointments.

Search for one or more appointments and link all matches to the process ticket.

<table>
<thead>
<tr>
<th>Appointment matching conditions</th>
<th>e.g., key: &quot;Title&quot;, value: &quot;some text&quot; or key: &quot;DynamicField_Name&quot;, value: &quot;Equalto1&quot;.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key: Limit</td>
<td>Value: 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Define the link type for found appointments.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link Type Normal</td>
</tr>
</tbody>
</table>

Fig. 13: Process Management Module LinkWithAppointment

**Appointment matching section**

This section is used to search for appointments.

Filters can be added with key-value pairs. There is logical AND relation between the filters if more than one filter is added. Logical OR relation can be added by multiple values separated by ,.

The key Limit limits the number of appointments returned. This field is mandatory.

**Link type section**

Here can be defined the link type for found appointments. Possible values are *Normal*, *Parent* or *Child*.

**See also:**

See the *LinkWithAppointment* and the *AppointmentSearch()* API reference.

**API Reference**

These API references are not available online, but they are included in this manual.

**AppointmentDataPull API**

<table>
<thead>
<tr>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernel::System::ProcessManagement::Modules::AppointmentDataPull - A module to fetch...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>All AppointmentDataPull functions.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PUBLIC INTERFACE</th>
</tr>
</thead>
<tbody>
<tr>
<td>new()</td>
</tr>
</tbody>
</table>

Don't use the constructor directly, use the ObjectManager instead:

```perl
my $AppointmentDataPullObject = $Kernel::OM->Get('Kernel::System::ProcessManagement::Modules::AppointmentDataPull');
```

(continues on next page)
Run()

Run Data

my $Success = $AppointmentDataPullObject->Run{
    UserID => 123,
    Ticket => \%Ticket,  # required
    ProcessEntityID => 'P123',
    ActivityEntityID => 'A123',
    SequenceFlowEntityID => 'T123',
    SequenceFlowActionEntityID => 'TA123',
    Config => {
        ConfigSearchKeyValueList {
            # Appointment title (optional)
            Title => '%SomeText1%,%SomeText2%',

            # Appointment location (optional)
            Location => '%SomeText1%,%SomeText2%',

            # Appointment description (optional)
            Description => '%SomeText1%,%SomeText2%'

            # Appointment calendars
            CalendarIDs => '1,2,3',
            Calendars => 'CalendarA,CalendarB',

            # Appointment teams
            TeamIDs => '1,2,3',
            Teams => 'TeamA,TeamB',

            # Appointment resources
            ResourceIDs => '1,2,3',

            # DynamicFields (optional) as single or comma separated values.
            DynamicField_FieldNameA => 'Empty=1',
            DynamicField_FieldNameB => 'Equals=123,456',
            DynamicField_FieldNameC => 'Like=value*',
            DynamicField_FieldNameD => 'GreaterThan=2001-01-01 01:01:01',
            DynamicField_FieldNameE => 'GreaterThanEquals=2001-01-01 01:01:01',
            DynamicField_FieldNameF => 'SmallerThan=2002-02-02 02:02:02',
            DynamicField_FieldNameG => 'SmallerThanEquals=2002-02-02 02:02:02',

            # AllDay appointments
            AllDay => 1,

            # Appointments created more than 60 minutes ago (appointment older than
            -> 60 minutes) (optional)
            AppointmentCreateTimeOlderMinutes => 60,

            # Appointments created less than 120 minutes ago (appointment newer than
            -> 120 minutes) (optional)
            AppointmentCreateTimeNewerMinutes => 120,

            # Appointments with create time after ... (appointment newer than this
            -> date) (optional)
            AppointmentCreateTimeNewerDate => '2006-01-09 00:00:01',
        }
    }
}
Appointments with created time before ... (appointment older than this date) (optional)
AppointmentCreateTimeOlderDate => '2006-01-19 23:59:59',

Appointments started more than 60 minutes ago (appointment start time older than 60 minutes) (optional)
AppointmentStartTimeOlderMinutes => 60,

Appointments started less than 120 minutes ago (appointment start time newer than 120 minutes) (optional)
AppointmentStartTimeNewerMinutes => 120,

Appointments with start time after ... (appointment start time newer than this date) (optional)
AppointmentStartTimeNewerDate => '2006-01-09 00:00:01',

Appointments ended more than 60 minutes ago (appointment end time older than 60 minutes) (optional)
AppointmentEndTimeOlderMinutes => 60,

Appointments ended less than 120 minutes ago (appointment end time newer than 120 minutes) (optional)
AppointmentEndTimeNewerMinutes => 120,

Appointments with end time after ... (appointment end time newer than this date) (optional)
AppointmentEndTimeNewerDate => '2006-01-09 00:00:01',

Appointments with end time before ... (appointment end time older than this date) (optional)
AppointmentEndTimeOlderDate => '2006-01-19 23:59:59',

Appointments which are inside the given period. Both values must be given for this filter. (optional)
AppointmentPeriodTimeNewerDate => '2006-01-09 00:00:01',
AppointmentPeriodTimeOlderDate => '2006-01-19 23:59:59',

}
# * To set process ticket title to be exactly the linked appointment title:
# Title => '<OTRS_APPOINTMENT_Title>',
# * To set the process ticket dynamic field NameX to an static value (not a real pull):
# DynamicField_NameX => 'someValue',
};

`Ticket` contains the result of TicketGet including DynamicFields.

AppointmentDataPush API

NAME

Kernel::System::ProcessManagement::Modules::AppointmentDataPush - A module to insert data to a linked appointment.

DESCRIPTION

All AppointmentDataPush functions.

PUBLIC INTERFACE

new()

Don't use the constructor directly, use the ObjectManager instead:

my $AppointmentDataPushObject = $Kernel::OM->Get(
  'Kernel::System::ProcessManagement::Modules::AppointmentDataPush');

Run()

Run Data

my $Success = $AppointmentDataPushObject->Run(
  UserID => 123,
  Ticket => \%Ticket, # required
  ProcessEntityID => 'P123',
  ActivityEntityID => 'A123',
  SequenceFlowEntityID => 'T123',
  SequenceFlowActionEntityID => 'TA123',
  Config => {
    ConfigSearchKeyValueList {
      # Appointment title (optional)
      Title => '%SomeText1%,%SomeText2%',
      # Appointment location (optional)
      Location => '%SomeText1%,%SomeText2%',
      # Appointment description (optional)
      Description => '%SomeText1%,%SomeText2%'
      # Appointment calendars
    }
  })

(continues on next page)
CalendarIDs => '1,2,3',
Calendars => 'CalendarA,CalendarB',

# Appointment teams
TeamIDs => '1,2,3',
Teams => 'TeamA,TeamB',

# Appointment resources
ResourceIDs => '1,2,3',

# DynamicFields (optional) as single or comma separated values.
DynamicField_FieldNameA => 'Empty=1',
DynamicField_FieldNameB => 'Equals=123,456',
DynamicField_FieldNameC => 'Like=value*',
DynamicField_FieldNameD => 'GreaterThan=2001-01-01 01:01:01',
DynamicField_FieldNameE => 'GreaterThanEquals=2001-01-01 01:01:01',
DynamicField_FieldNameF => 'SmallerThan=2002-02-02 02:02:02',
DynamicField_FieldNameG => 'SmallerThanEquals=2002-02-02 02:02:02',

# AllDay appointments
AllDay => 1,

# Appointments created more than 60 minutes ago (appointment older than...
AppointmentCreateTimeOlderMinutes => 60,

# Appointments created less than 120 minutes ago (appointment newer than...
AppointmentCreateTimeNewerMinutes => 120,

# Appointments with create time after ... (appointment newer than this...
AppointmentCreateTimeNewerDate => '2006-01-09 00:00:01',

# Appointments with created time before ... (appointment older than this...
AppointmentCreateTimeOlderDate => '2006-01-19 23:59:59',

# Appointments started more than 60 minutes ago (appointment start time...
AppointmentStartTimeOlderMinutes => 60,

# Appointments started less than 120 minutes ago (appointment start time...
AppointmentStartTimeNewerMinutes => 120,

# Appointments with start time after ... (appointment start time newer...
AppointmentStartTimeNewerDate => '2006-01-09 00:00:01',

# Appointments with start time before ... (appointment start time older...
AppointmentStartTimeOlderDate => '2006-01-19 23:59:59',

# Appointments ended more than 60 minutes ago (appointment end time older...
AppointmentEndTimeOlderMinutes => 60,
# Appointments ended less than 120 minutes ago (appointment end time newer than 120 minutes) (optional)
AppointmentEndTimeNewerMinutes => 120,

# Appointments with end time after ... (appointment end time newer than this date) (optional)
AppointmentEndTimeNewerDate => '2006-01-09 00:00:01,'

# Appointments with end time before ... (appointment end time older than this date) (optional)
AppointmentEndTimeOlderDate => '2006-01-19 23:59:59',

# Appointments which are inside the given period. Both values must be given for this filter. (optional)
AppointmentPeriodTimeNewerDate => '2006-01-09 00:00:01',
AppointmentPeriodTimeOlderDate => '2006-01-19 23:59:59',

UserID => 123,  # optional, to override the UserID from the logged user

# Value set:
# * Key is the attribute of the linked appointments where the data will be pushed
# * Value is the value to be set, supporting smart tags <OTRS_TICKET_*>
# from the current process ticket e.g.
# # Example:
# # * To set linked appointments title to be exactly the process ticket title:
# Title => '<OTRS_Ticket_Title>',
# # * To set the location name to an static value:
# Location => 'Meeting Room',

};

- `Ticket` contains the result of TicketGet including DynamicFields.

**LinkWithAppointment API**

**NAME**
Kernel::System::ProcessManagement::Modules::LinkWithAppointment - A module to link appointments.

**DESCRIPTION**
All LinkWithAppointment functions.

**PUBLIC INTERFACE**
new()

Don't use the constructor directly, use the ObjectManager instead:
my $LinkWithAppointmentObject = $Kernel::OM->Get('Kernel::System::ProcessManagement::Modules::LinkWithAppointment');

Run()

Run Data

my $Success = $LinkWithAppointmentObject->Run(

UserID => 123,
Ticket => \%Ticket, # required
ProcessEntityID => 'P123',
ActivityEntityID => 'A123',
SequenceFlowEntityID => 'T123',
SequenceFlowActionEntityID => 'TA123',
Config => {

    ConfigSearchKeyValueList {
        AppointmentID => 1234,
        AppointmentID => [1234, 1235],

        # Appointment title (optional)
        Title => '%SomeText1%,%SomeText2%',

        # Appointment location (optional)
        Location => '%SomeText1%,%SomeText2%',

        # Appointment description (optional)
        Description => '%SomeText1%,%SomeText2%',

        # Appointment calendars
        CalendarIDs => '1,2,3',
        Calendars => 'CalendarA,CalendarB',

        # Appointment teams
        TeamIDs => '1,2,3',
        Teams => 'TeamA,TeamB',

        # Appointment resources
        ResourceIDs => '1,2,3',

        # DynamicFields (optional) as single or comma separated values.
        DynamicField_FieldNameA => 'Empty=1',
        DynamicField_FieldNameB => 'Equals=123,456',
        DynamicField_FieldNameC => 'Like=value*',
        DynamicField_FieldNameD => 'GreaterThan=2001-01-01 01:01:01',
        DynamicField_FieldNameE => 'GreaterThanEquals=2001-01-01 01:01:01',
        DynamicField_FieldNameF => 'SmallerThan=2002-02-02 02:02:02',
        DynamicField_FieldNameG => 'SmallerThanEquals=2002-02-02 02:02:02',

        # AllDay appointments
        AllDay => 1,

        # Appointments created more than 60 minutes ago (appointment older than 60 minutes) (optional)
        AppointmentCreateTimeOlderMinutes => 60,

        # Appointments created less than 120 minutes ago (appointment newer than...
# Appointments with create time after ... (appointment newer than this date) (optional)
AppointmentCreateTimeNewerDate => '2006-01-09 00:00:01',

# Appointments with created time before ... (appointment older than this date) (optional)
AppointmentCreateTimeOlderDate => '2006-01-19 23:59:59',

# Appointments started more than 60 minutes ago (appointment start time older than 60 minutes) (optional)
AppointmentStartTimeOlderMinutes => 60,

# Appointments started less than 120 minutes ago (appointment start time newer than 120 minutes) (optional)
AppointmentStartTimeNewerMinutes => 120,

# Appointments with start time after ... (appointment start time newer than this date) (optional)
AppointmentStartTimeNewerDate => '2006-01-09 00:00:01',

# Appointments with start time before ... (appointment start time older than this date) (optional)
AppointmentStartTimeOlderDate => '2006-01-19 23:59:59',

# Appointments ended more than 60 minutes ago (appointment end time older than 60 minutes) (optional)
AppointmentEndTimeOlderMinutes => 60,

# Appointments ended less than 120 minutes ago (appointment end time newer than 120 minutes) (optional)
AppointmentEndTimeNewerMinutes => 120,

# Appointments with end time after ... (appointment end time newer than this date) (optional)
AppointmentEndTimeNewerDate => '2006-01-09 00:00:01',

# Appointments with end time before ... (appointment end time older than this date) (optional)
AppointmentEndTimeOlderDate => '2006-01-19 23:59:59',

# Appointments which are inside the given period. Both values must be given for this filter. (optional)
AppointmentPeriodTimeNewerDate => '2006-01-09 00:00:01',
AppointmentPeriodTimeOlderDate => '2006-01-19 23:59:59',

ConfigDropdown => {
    Behavior => 'NoCopy', # 'NoCopy', 'CopyFirstLinked' or 'CopyLastLinked' only
    UserID => 123, # optional, to override the UserID from the logged user
}
12.2 Agent Interface

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

12.2.1 Resource Calendar

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

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

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

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

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

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

Calendar Widget Configuration

The widget configuration can be accessed via the gear icon in top right corner of any calendar widget.

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

The following settings are available when adding or editing this resource. The fields marked with an asterisk are mandatory.
Fig. 14: Resource Calendar Widget
Fig. 15: Resource Calendar Settings
General

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

Grid View *
Here you can select the time period for the grid view.

Timeline View *
Here you can select the time period for the timeline view. You can also define a custom timeline view here. If you have selected the Custom timeline view, you can configure the settings for this timeline in the Timeline View "Custom" Settings section below.

Agenda View *
Here you can select the time period for the agenda view.

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

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

See also:
The additional resource information can be added in setting AppointmentCalendar::CustomTimeline::ResourceInformation in the system configuration.

Resources

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

Timeline View “Custom” Settings

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

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

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

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

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

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

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

![Resource]

Team

Agent

Fig. 16: Add Appointment Action

Calendar Subscriptions

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

To use the calendar subscription feature:

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

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

12.2.2 Statistics and Reports

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

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

- Age
- Days
• **Hours**

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

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

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

### 12.3 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 implements a category folder structure for standard templates.

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

- SILVER, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager

- OTRSCategoriesForTextModules

13.1 Administrator Interface

After installation of the package two new modules will be available in the Ticket Settings group of the administrator interface.
13.1.1 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.

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 cannot 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.
3. Click on the **Save or Save and finish** button.

![Edit Category Screen](image)

**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 uppercase letters and spaces. The name will be displayed in the overview table.

**Sub-category of**

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

**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.

### 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.

**Queue**

The queue where the ticket should be placed to use the template.
Add Template

- **Type**: Answer
- **Name**: 
- **Subject**: 

Subject method:

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

**Template:**

- Format:
- Font:
- Size:
- Text:
- Source:

**To**: 
**Cc**: 
**Bcc**: 
**Queue**: 
**Category**: 
**Attachments**: 

- **Validity**: valid

**Comment**: 

**Save** or **Cancel**

Fig. 4: Add Template Screen
Category

The category for the text template. The category can be also assigned in the Templates ↔ Categories screen.

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.

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.

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.

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.

### 13.2 Agent Interface

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

#### 13.2.1 Articles

This package implements a category folder structure for standard templates. This folder structure will be shown in any forms that produce an article. The folder structure is a new drop-down field named *Text Template by Category*, where the template can be selected in a category tree.

**Note:** The type of the template and the queue of the ticket determine whether a template is displayed in a specific category folder structure. The category determines in which category structure the template is shown.
13.3 External Interface

This package has no external interface.
CI ASSIGNMENT ATTRIBUTE DYNAMIC FIELD MAP

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

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

Benefits

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

Target Groups

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

Available in Service Package

• SILVER, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager

• OTRSCIAssignmentAttributeDynamicFieldMap

Note: This feature add-on requires the Configuration Management feature.
14.1 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.

14.1.1 Assign Service and SLA

It is possible to add definition of two configurable fields in configuration items which contain service and SLA information. Whenever a configuration item is linked to a ticket or unlinked from a ticket, these fields are used to update or remove service and SLA of the linked ticket.

To add the fields for configuration items:

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

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

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

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

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

Requirements and limitations for service and SLA linking:

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

Requirements and limitations for service and SLA unlinking:

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

### 14.1.2 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 the 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:

<table>
<thead>
<tr>
<th>Configuration Item Attribute</th>
<th>Dynamic Field Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIC::IPAddress</td>
<td>IPAddress</td>
</tr>
<tr>
<td>NIC::IPoverDHCP</td>
<td>IPoverDHCP</td>
</tr>
<tr>
<td>SerialNumber</td>
<td>SerialNumber</td>
</tr>
<tr>
<td>WarrantyExpirationDate</td>
<td>WarrantyExpirationDate</td>
</tr>
</tbody>
</table>

   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:

<table>
<thead>
<tr>
<th>Configuration Item Attribute</th>
<th>Dynamic Field Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIC::IPoverDHCP</td>
<td>Yes → 1</td>
</tr>
<tr>
<td></td>
<td>No → 2</td>
</tr>
</tbody>
</table>

   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 can be shown in the Properties widget of the ticket detail view. The dynamic fields 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.

14.2 Agent Interface

This package has no agent interface.

14.3 External Interface

This package has no external interface.
With this feature add-on it is possible to enhance the configuration items in the configuration management database (CMDB) through additional fields and to create links and references between these or other data in OTRS. This can be useful for linking configuration items with each other when there is a dependency relationship between them or to store information regarding linked services and agents directly in the configuration item, as well as find this information with a quicker autocomplete search.

The following input fields with autocomplete search can be added:

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

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

**Benefits**

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

**Target Groups**

- Configuration item managers
- Internal and external IT
- Facility management
- Sales

**Available in Service Package**

- SILVER, GOLD, TITANIUM, PLATINUM

**Package Name in OTRS Package Manager**

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

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 agents.

To add the fields for configuration items:

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

The following sections describe the possible input fields.

ReferenceCI Field

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

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

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

**Key** *

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

**Name** *

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

Note: It is recommended to always use English words for names.
See also:

Names can be translated into other languages with custom language files. For more information, see the Custom Language File chapter in the administration 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:

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

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

**Key** *

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

**Name** *

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

---

**Note:** It is recommended to always use English words for names.

---

**See also:**

Names can be translated into other languages with custom language files. For more information, see the Custom Language File chapter in the administration 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 Reference-Service 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 agents 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 language files. For more information, see the Custom Language File chapter in the administration 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 agent 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.

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

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

To add a configuration item:

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

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

See also:
The fields in the Create Configuration Item screen can be very different on each classes. The new fields added by this feature add-on is described in the Administrator Interface chapter.

15.3 External Interface

This package has no external interface.
CHAPTER SIXTEEN

CIS IN CUSTOMER FRONTEND

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

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

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

Benefits

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

Target Groups

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

Available in Service Package

- SILVER, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager

- OTRSCIsInCustomerFrontend

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

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

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

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

    ![Text Field](https://example.com)

    **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.

    ![CustomerCompany Field](https://example.com)

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. Add the endpoint to *External Interface*.

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

2. Select a configuration item class and click on the *Change class definition* button.

3. Add the attribute `NotForCustomer: 1` to the field definition.

For example:
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.

**Key:** Vendor  
**Name:** Vendor  
**Searchable:** 1  
**Input:**  
  **Type:** Text  
  **Size:** 50  
  **MaxLength:** 50  
  **NotForCustomer:** 1
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
See also:

The visible columns can be defined in the following setting:

- `ExternalFrontend::ITSMConfigItemOverview###ShowColumns`

If multiple classes of configuration items are available in the external interface, an *All* filter is displayed to see all classes. The *Class* column is always visible, when the filter *All* is selected.

Clicking on a configuration item will show the detail view.
### Configuration Item Details

<table>
<thead>
<tr>
<th>Customer Company</th>
<th>Super Support Inc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>&quot;Lacey Green&quot; <a href="mailto:lacey@example.org">lacey@example.org</a></td>
</tr>
<tr>
<td>CPU</td>
<td></td>
</tr>
<tr>
<td>Ram</td>
<td></td>
</tr>
<tr>
<td>Hard Disk</td>
<td></td>
</tr>
<tr>
<td>Capacity</td>
<td></td>
</tr>
<tr>
<td>Network Adapter</td>
<td>NIC</td>
</tr>
<tr>
<td>IP over DHCP</td>
<td>Yes</td>
</tr>
</tbody>
</table>

![Configuration Item Details Screen](image)

Fig. 2: Configuration Item Details Screen

16.3. External Interface
CHAPTER
SEVENTEEN

CONFIGURATION MANAGEMENT CONNECTOR

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

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

Available in Service Package
- SILVER, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager
- OTRSConfigurationManagementConnector

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

17.1 Administrator Interface

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

17.1.1 Processes & Automation

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

Web Services

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

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

To use this invoker:

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

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

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

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

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

The configuration item Number attribute is used to make the logical connection between the remote data and the data stored in the OTRS database. To create and update configuration items with this invoker values for the following keys are required:
Mapping ITSM::ConfigItem

Default rule for unmapped keys: **Keep (leave unchanged)**
This rule will apply for all keys with no mapping rule.

Default rule for unmapped values: **Keep (leave unchanged)**
This rule will apply for all values with no mapping rule.

New static value:

Key: **Class**  Value: **Computer**

New key map:

Save or Save and finish or Cancel

---

Fig. 2: Static Mapping for ITSM::ConfigItem

Mapping for Key Interface::###:NDEX###:IPAddress

Key mapping:

- Map key: **IPList**
- matching the: **Exact value(s)**
- to new
  - key: **DEX###:IPAddress**

List separator:

Value mapping:

New value map:

---

Fig. 3: Key Mapping for ITSM::ConfigItem

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

### Generic Link Object Invoker

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

To use this invoker:

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

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

- SourceClass
- SourceObject
- TargetClass
- TargetObject
- Type

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

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

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

### Configuration Item Synchronization Invoker

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

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

To use this invoker:

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

#### Default rule for unmapped keys:
- **Keep (leave unchanged)**
  - This rule will apply for all keys with no mapping rule.

#### Default rule for unmapped values:
- **Keep (leave unchanged)**
  - This rule will apply for all values with no mapping rule.

#### New static value:

- **Key:** SourceClass
- **Value:** Computer

- **Key:** SourceObject
- **Value:** ITSMConfigItem

- **Key:** TargetClass
- **Value:** Software

- **Key:** TargetObject
- **Value:** ITSMConfigItem

- **Key:** Type
- **Value:** DependOn

#### New key map:

---

#### Mapping for Key SourceNumber

- **Key mapping:**
  - **Map key:** OTRS@ComputerN
  - **key:** SourceNumber

- **List separator:**

- **Value mapping:**

#### Mapping for Key TargetNumber

- **Key mapping:**
  - **Map key:** OTRS@SoftwareN
  - **key:** TargetNumber

- **List separator:**

- **Value mapping:**

---

**Fig. 4: Mapping Link**

---

**17.1. Administrator Interface**
It works with two steps on every sync action:

1. At first depending on the change that was made in the local system (add, update or delete a configuration item) the invoker performs a related search on the remote system to determine, if it is needed to create, update or delete a remote configuration item.

2. On the second step, a related invoker for create actions, update actions or delete actions is called, which executes the main changes on the remote system. Those invokers have to be created as well and needs to be of invoker type `ITSM::ConfigItem` to make sure the communication works properly.

See the enclosed overview of example invokers:

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

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

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

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

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

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

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

This package has no agent interface.

17.3 External Interface

This package has no external interface.
Fig. 7: Search Invoker Details
Mapping ITSMConfigItem

Default rule for unmapped keys: Keep (leave unchanged)
This rule will apply for all keys with no mapping rule.

Default rule for unmapped values: Keep (leave unchanged)
This rule will apply for all values with no mapping rule.

New static value: 

- Key: Password Value: MyPassword
- Key: UserLogin Value: MyUserLogin

New key map: 

Save or Save and finish or Cancel

Fig. 8: Search Invoker Mapping

Operations
Operations are individual system functions which remote systems can request.

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
<th>CONTROLLER</th>
<th>INBOUND MAPPING</th>
<th>OUTBOUND MAPPING</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConfigItemCreate</td>
<td>-</td>
<td>ConfigItem::ConfigItemCreate</td>
<td>ITSMConfigItem</td>
<td>ITSMConfigItem</td>
</tr>
<tr>
<td>ConfigItemDelete</td>
<td>-</td>
<td>ConfigItem::ConfigItemDelete</td>
<td>ITSMConfigItem</td>
<td>ITSMConfigItem</td>
</tr>
<tr>
<td>ConfigItemSearch</td>
<td>-</td>
<td>ConfigItem::ConfigItemSearch</td>
<td>ITSMConfigItem</td>
<td>ITSMConfigItem</td>
</tr>
<tr>
<td>ConfigItemUpdate</td>
<td>-</td>
<td>ConfigItem::ConfigItemUpdate</td>
<td>ITSMConfigItem</td>
<td>ITSMConfigItem</td>
</tr>
</tbody>
</table>

Fig. 9: Operation Overview
CHAPTER
EIGHTEEN

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, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager
- OTRSCreditCardFilter

18.1 Administrator Interface

This package has no administrator interface.
18.1.1 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.

**Note:** This feature is only available to *On-Premise* customers. If you are a *Managed* customer, this feature is taken care of by the *Customer Solutions Team* in OTRS. Please contact us via support@otrs.com or in the OTRS Portal.

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.

<table>
<thead>
<tr>
<th>Issuing Network, Card Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>JCB, 3528988095245935</td>
</tr>
<tr>
<td>JCB, 3112606824580636</td>
</tr>
<tr>
<td>JCB, 3096030869937728</td>
</tr>
<tr>
<td>JCB, 3112437499296450</td>
</tr>
<tr>
<td>JCB, 3096010732100407</td>
</tr>
<tr>
<td>JCB, 3528461498782367</td>
</tr>
<tr>
<td>JCB, 3112892137191440</td>
</tr>
<tr>
<td>JCB, 3088814635323630</td>
</tr>
</tbody>
</table>

3. Open the ticket detail view 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:

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

5. Refresh the ticket detail view. The article the body will shown as:

<table>
<thead>
<tr>
<th>Issuing Network, Card Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>JCB, 352898xxxxxx5935</td>
</tr>
<tr>
<td>JCB, 311260xxxxxx0636</td>
</tr>
<tr>
<td>JCB, 309603xxxxxx7728</td>
</tr>
<tr>
<td>JCB, 311243xxxxxx6450</td>
</tr>
<tr>
<td>JCB, 309601xxxxxx0407</td>
</tr>
<tr>
<td>JCB, 352846xxxxxx2367</td>
</tr>
<tr>
<td>JCB, 311289xxxxxx1440</td>
</tr>
<tr>
<td>JCB, 308881xxxxxx3630</td>
</tr>
</tbody>
</table>

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:

```bash
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.

## 18.2 Agent Interface

This package has no agent interface.

### 18.2.1 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 4-4-4-1 (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.

### 18.2.2 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,18007968437698
   JCB 15 digit,18000143488683
   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 detail view 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.

**18.2.3 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:
5. Open the ticket detail view. The warning message should appear next to the credit card number.

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.

### 18.3 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 subcontractors, as it adds, in addition to the customer detail view, a second possibility for storing customer data in a ticket and including it in communication processes.

Benefits

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

Target Groups

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

Available in Service Package

- SILVER, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager

- OTRSCustomContactFields
19.1 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.

19.1.1 Communication & Notifications

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

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

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

Ticket Notifications

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

Sending Articles Created in the External Interface as Emails

To activate sending emails to custom contact field customer users:

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

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

   - Select the checkbox in the *Article visible to customer* field.
8. In the *Notification Text* section:
   - If you want the contacts of the custom contact field to get the title of the article as mail subject add `<OTRS_CUSTOMER_SUBJECT>` into the *Subject* field.
• If you want to send the article body as mail body add \textlt{\texttt{<OTRS\_CUSTOMER\_BODY>}} into the Text field.

9. Click on the Save button.

19.1.2 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.

![Customer Field Settings](image)

Fig. 1: Customer Dynamic Field Settings
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.

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 Fisname from the Filter contact by has to be selected. After that the Filter criteria field will be displayed. Now we can insert Theo into the filter field. Currently it is not possible to define a regular expression or a placeholder for filtering customer users.

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

19.2 Agent Interface

This package has no agent interface.
19.2.1 New Features

**Storing/changing customer user in a Customer field**

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

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

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

**Showing Customer dynamic field values for a ticket**

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

**Using Customer dynamic field values for communication in a ticket**

If a field has been configured to use the contacts for communication and the *Reply via Email* or *Reply to All via Email* actions are used in the ticket detail view, all customer users stored in the respective field will be added to the configured recipient list (to, cc or bcc) unless they exist as a recipient already.

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

**Searching tickets for Customer dynamic field values**

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

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

19.3 External Interface

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

![Fig. 2: Custom Contact Ticket List](image)
If a dynamic field of type *Customer* contains the customer user who is logged in to the external interface, the *Ticket List* screen is extended with a new menu item. This menu item lists the tickets, where the customer user is added as dynamic field. The label of this menu item will be the label of the dynamic field.

This feature allows you to list tickets for customer users that they do not otherwise have access to. The customer users can be added to any ticket as dynamic field, even if they are not the customer user of the ticket.
CHAPTER TWENTY

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, GOLD, TITANIUM, PLATINIUM

Package Name in OTRS Package Manager

- OTRSCustomerActivityWidget

20.1 Administrator Interface

This package has no administrator interface.

20.2 Agent Interface

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

Every agent can choose the customers he wants to see via a dashboard widget. After installation of the package the widget is empty, since no customer has been selected yet.

![Fig. 1: Customer Activity Widget Without Selected Customer](image)

Clicking on the Add Customer button opens an action where you can filter for one or more desired customers and select them via a checkbox. Submitting the form will populate the widget with the customers.

**Note:** If an agent is already using the widget in OTRS 7, the settings and the content will be fully migrated from the previous version.

![Fig. 2: Customer Activity Widget](image)

For the selected customers the number of open/unlocked, open/locked and escalated tickets will be shown. The tickets are summed per row and per column. Clicking on the customer itself leads to the customer detail view. Clicking on one of the number in each row leads to a separate ticket list, where exactly these tickets are displayed. If you hover over the numbers in the matrix instead, a popover card with exactly the same list of tickets will be displayed, so you can have a quick look at the tickets behind the numbers. This does not apply to the assigned numbers in the Total row at the footer of the widget.

To turn on and off the widget:

1. Open the screen configuration of the Dashboard.
2. Add or remove the Customer Activity widget. You can rearrange the widgets as well.
3. Click on the Save button.

To add customers to the widget:

1. Click on the plus button in the header of the widget.
2. Select the customers that you want to display in the widget.
3. Click on the Add Selected button.

To remove a customer from the widget:

1. Click on the \(\times\) button in the last column.
2. Click on the Yes button in the confirmation dialog.

The behavior of the ticket count feature can be changed in the widget configuration.

```
Fig. 3: Customer Activity Widget Configuration

Count Tickets *
If you select All tickets, every ticket will be counted, even those tickets for which the agent does not have the sufficient rights. Selecting the Permitted tickets option will count only the tickets for which the agent has proper permissions.

Ticket Permissions *
Controls if read only or read-write permission is needed to count the tickets.
```

20.3 External Interface

This package has no external interface.
CUSTOMER EVENT TICKET CALENDAR

This feature add-on 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 an online meeting.

Benefits

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

Target Groups

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

Available in Service Package

- SILVER, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager

- OTRSCustomerEventTicketCalendar

21.1 Administrator Interface

This package has no administrator interface.

21.2 Agent Interface

Each appointment in a calendar has got a new Visibility field. This field is mandatory and can only be set by an agent. The Visibility field has the following options:

External Restricted

The customer user needs to login in the external interface to see the appointment.

Internal

This is the default selection for new appointments that do not show any data of the associated tickets to the customer user. In existing appointments where no visibility value has set is considered also internal and the customer user cannot view the appointment.

Public

Every visitor of the external interface can see the appointment.
21.3 External Interface

The external interface has a new menu item to show appointments. In this screen all appointments are displayed that have the visibility set to External Restricted by an agent.

![Fig. 1: Appointments Screen in External Interface](image)

This screen has three tabs:

**Personal**

Appointments that have no linked tickets or the customer user is the assigned customer user of the ticket linked to the appointment.

**Company**

Appointments that have linked tickets where the primary or secondary customer of the customer user is the same as the customer of the logged in customer user.

**All**

Personal and company appointments.

Each tab has filters to selecting the time range for the displayed appointments. The possible values are Day, Week, Month and Year. Clicking on the Today button will show the appointments for the current day.

You can use the search box to filter the displayed appointments.

Hovering an appointment will give a brief overview about the appointment while clicking on an appointment will lead to the detail view of the appointment.

**See also:**

The field Resource is configurable in the system configuration to be shown or not. It is disabled by default.
CHAPTER TWENTYTWO

CUSTOMER FRONTEND LINK OBJECT

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

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

Benefits

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

Target Groups

- Internal and external IT
- Customer service
- Facility management

Available in Service Package

- SILVER, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager

- OTRSCustomerFrontendLinkObject

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

- CIs in Customer Frontend
22.1 Administrator Interface

This package has no administrator interface.

22.2 Agent Interface

This package has no agent interface.

To use this functionality on External Interface:

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

22.3 External Interface

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

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

To see the linked object in external interface:

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

Clicking on a ticket or knowledge base article will lead you to the linked ticket or knowledge base article.
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, GOLD, TITANIUM, PLATINUM

**Package Name in OTRS Package Manager**

- OTRSCustomerSpecificServices
23.1 Administrator Interface

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

23.1.1 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.

![Manage Customer-Service Relations](image)

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.

**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.

**Note:** This feature can be affected if the setting CustomerUserExcludePrimaryCustomerID is set to 1 in the Kernel/Config.pm. By excluding the primary customer ID and not setting up customer IDs manually, no services are assigned.
will be shown.

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](image)

**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](image)

**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.

![Allocate Services to Customer Screen](image)

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.

23.2 Agent Interface

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

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

23.2.1 Customers

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

![Services Widget](image)

Fig. 5: Services Widget

23.3 External Interface

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

23.3.1 Restrict Service Selection

See also:

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

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

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

Benefits

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

Target Groups

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

Available in Service Package

- SILVER, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager

- OTRSDashboardNewsWidget

24.1 Administrator Interface

This chapter describes the new features that are available in the administrator interface after installation of the package.
24.1.1 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 Screen](image1)

**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 Screen](image2)

**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.

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.

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.

![Edit Announcement Screen with Rich Text Editor](image)

**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.
## 24.2 Agent Interface

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

![Fig. 6: Announcements Widget](image)

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

![Fig. 7: Announcement Details](image)

## 24.3 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**

- SILVER, GOLD, TITANIUM, PLATINUM

**Package Name in OTRS Package Manager**

- OTRSDataPrivacyProtection

**Note:** This feature is available only for managed customers.
25.1 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.

Note: This feature is only available to On-Premise customers. If you are a Managed customer, this feature is taken care of by the Customer Solutions Team in OTRS. Please contact us via support@otrs.com or in the OTRS Portal.

25.1.1 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*.

### 25.1.2 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:

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

### 25.1.3 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 `Dynamic-Field_` and the related field name.

Possible object filters:

- ValidID
- CustomerID
- CustomerCompanyStreet
- CustomerCompanyURL
- CustomerCompanyComment

(continues on next page)
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:

```yaml
---
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:

```yaml
---
RuleName: Delete customer company name and customer company country by customer company name with wildcard search.
RuleType: PrivacyByDeletion
RuleSource: someRuleSource
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 `Dynamic-Field_` 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 (>=) 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 `User-CustomerID` with `company` will in fact search for `*company*`. 

174 Chapter 25. Data Privacy Protection
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:

```yaml
---
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:

```yaml
---
RuleName: Anonymize user first names and user last names by user first name and without wildcard search.
RuleType: PrivacyByAnonymization
RuleSource: someRuleSource
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:

```yaml
---
RuleName: Delete user first names and user last names by user first name and user last name with wildcard search.
RuleType: PrivacyByDeletion
RuleSource: someRuleSource
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

(continues on next page)
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
- StateTypeID
- Priorities
- PriorityIDs
- Services
- ServiceIDs
- SLAs
- SLAID
- Locks
- LockIDs
- OwnerIDs
- ResponsibleIDs

(continues on next page)
- 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
- TicketChangeTimeNewerDate
- TicketChangeTimeOlderDate
- TicketLastChangeTimeOlderMinutes
- TicketLastChangeTimeNewerMinutes
- TicketLastChangeTimeNewerDate
- TicketLastChangeTimeOlderDate
- TicketChangeTimeOlderDate
- TicketCloseTimeOlderMinutes
- TicketCloseTimeNewerMinutes
- TicketCloseTimeNewerDate
- TicketCloseTimeOlderDate
- TicketPendingTimeOlderMinutes
- TicketPendingTimeNewerMinutes
- TicketPendingTimeNewerDate
- TicketPendingTimeOlderDate
- TicketEscalationTimeOlderMinutes
- TicketEscalationTimeNewerMinutes
- TicketEscalationTimeNewerDate
- TicketEscalationTimeOlderDate
- TicketEscalationUpdateTimeOlderMinutes
- TicketEscalationUpdateTimeNewerMinutes
- TicketEscalationUpdateTimeNewerDate
- TicketEscalationUpdateTimeOlderDate
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:

<table>
<thead>
<tr>
<th>TicketNumber: 123546</th>
</tr>
</thead>
<tbody>
<tr>
<td>TicketNumber: 123546</td>
</tr>
<tr>
<td>TicketNumber: 123666</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title: SomeText</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title: SomeTest1</td>
</tr>
<tr>
<td>Title: SomeTest2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>States:</th>
</tr>
</thead>
<tbody>
<tr>
<td>new</td>
</tr>
<tr>
<td>open</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>StateIDs:</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
</tbody>
</table>

The corresponding YAML code could look as follows:

```yaml
RuleName: My Explanation Rule
RuleType: PrivacyByDeletion
RuleSource: GDPR
DataClassification: Ticket:
  - CustomerUserID
  - CustomerID
ObjectFilter:
  Ticket:
    Queue:
      - Junk
      - 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
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:

```yaml
---
RuleName: Delete ticket titles by state names, that are older than one month.
RuleSource: GDPR
RuleType: deletion
DataClassification:
  Ticket:
    - Title
ObjectFilter:
  Ticket:
    States:
    - new
    - open
    TicketCreateTimeOlderMinutes: 43200
```

Delete article subject and body by state names, that are located in specific queues:

```yaml
---
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:
    States:
    - new
    - open
    Queues:
```

(continues on next page)
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:
    - 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:
Object filter descriptions:

- **Limit**: Limits the number of search results.
- **CreateTime**: Searches for dates greater than or equal to (>=) the given time.
- **Valid**: Searches for valid or invalid users. Possible values are 0 or 1.
- **WildcardSearch**: Affects the object filters `UserFirstname`, `UserLastname`, `UserLogin` and `UserTitle`. It will wrap the search term in wildcards, so they match all objects that contain the supplied value. For example, if this option is set to 1, filtering on `UserLogin` with `agent` will in fact search for `*agent*`.
- **UserPreferences**: Array containing the user preferences like user email address as keys with certain search criteria as values (see YAML configuration examples).

**Rule Configuration Examples**

Here are some examples for rule configurations. These examples are valid YAML codes. You can copy these examples and modify them according to your needs.

Delete user first names by user first name:

```yaml
---
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:

```yaml
---
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:**  
  - User  
  - CreateTime: 2019-01-01  
  - UserFirstname: someFirstname

## 25.2 Agent Interface

This package has no agent interface.

## 25.3 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, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager
- OTRSDeleteAttachments

26.1 Administrator Interface

This package has no administrator interface.

Note: This feature is only available to On-Premise customers. If you are a Managed customer, this feature is taken care of by the Customer Solutions Team in OTRS. Please contact us via support@otrs.com or in the OTRS Portal.
26.1.1 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](image)

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:

**OTRSD DeleteAttachments::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.

**OTRSD DeleteAttachments::TimeSpan**

The starting time (in days) from the current date from which the tickets should be analyzed for deletion.

**OTRSD DeleteAttachments::TimePeriod**

The end time (in days) from the current date up to which the tickets should be analyzed for deletion.

Example:

```
<table>
<thead>
<tr>
<th>TimePeriod (40)</th>
<th>TimeSpan (10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>50 days ago</td>
<td>40 days ago</td>
</tr>
<tr>
<td>40 days ago</td>
<td>30 days ago</td>
</tr>
<tr>
<td>30 days ago</td>
<td>20 days ago</td>
</tr>
<tr>
<td>20 days ago</td>
<td>10 days ago</td>
</tr>
<tr>
<td>10 days ago</td>
<td>Today</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Ticket close time</td>
<td></td>
</tr>
</tbody>
</table>
```

Means that all tickets between 10 days ago from the current date and 40 days ago from the current date are analyzed for deletion.

**Note:** The value of `TimePeriod` must be always higher than the value of `TimeSpan`.

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###OTRSD DeleteAttachments**

Delete attachments from tickets.

**Daemon::SchedulerCronTaskManager::Task###OTRSD DeleteAttachments_AddMIME Types**

Populate system configuration MIME types for attachment deletion.
Note: Do not forget to restart the OTRS daemon after the system configuration is changed.

26.2 Agent Interface

This package has no agent interface.

26.3 External Interface

This package has no external interface.
CHAPTER
TWENTYSEVEN

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
- SILVER, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager
- OTRSDynamicFieldAttachment

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

27.1 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.
27.1.1 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.

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.

### 27.2 Agent Interface

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

To upload files to dynamic field of type attachment:

1. Make sure that the dynamic field of type attachment is added to the desired screen.
2. Click on the uploading area or just dropping the file there. The file gets uploaded immediately to the server and a new upload field will be displayed as long as the configured maximum amount of attachments has not been reached.
3. Click on the Send button. If a file has exceeded the maximum attachment size, a message will appear warning the agent that the file exceeds the maximum attachment size. In this case the file will not be uploaded and the ticket will be created without problem.

**Documents**

![Drop files here or click to select files](image)

<table>
<thead>
<tr>
<th>Contract.pdf</th>
<th>Document</th>
<th>25.28 KB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Privacy.txt</td>
<td>Text</td>
<td>15.44 KB</td>
</tr>
<tr>
<td>Terms.docx</td>
<td>Image</td>
<td>119.94 KB</td>
</tr>
</tbody>
</table>

**Fig. 2: Dynamic Field Attachment in Action Form**

The dynamic field of type attachment is displayed in the ticket detail view or in the header of the article, if it is added to the screens and at least one file has been uploaded.

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

1. Find the dynamic field in the Properties widget of the ticket detail view.
2. Click on the download icon next to the filename to download the file.

The widget can be expanded to look similar to the Attachments widget.

If an image file is uploaded, it is possible to see a preview of the image.

To download a file from an article dynamic field of type attachment:
1. Select an article, that has a dynamic field of type attachment with at least one uploaded file.
2. Expand the article.
3. Find the attachments in the article header.
4. Click on a filename to download the file.

If a ticket dynamic field of type attachment has been configured to be displayed in a ticket action form, it is possible to delete the uploaded files and add new files.

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

1. Make sure that the dynamic field of type attachment is added to the Change Free Fields action.
2. Click on the X icon in the last column of an attachment. The attachment will be deleted without confirmation!

**Note:** It is not possible to delete files from dynamic field of type attachment created for article object.

Dynamic field of type attachment supports searching for the file names. To search for a file name of a dynamic field of type attachment, add the dynamic field to the search engine.
27.3 External Interface

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

- SILVER, GOLD, TITANIUM, PLATINUM

**Package Name in OTRS Package Manager**

- OTRSDynamicFieldCalculation

### 28.1 Administrator Interface

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

### 28.1.1 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](image)

**Calculation formula**

This setting is used to add a calculation formula. Simple mathematical operations are possible. The default value is zero.

Numbers in the formula should have dot as decimal separator and should be used without thousands separator like 123456.78.

**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 calculation can be added to many screens.

28.3 External Interface

This package has no external interface.
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
- Customer user ID
- Attribute

In addition, the following configurations are possible:

- Automatic linking and type of linking between configuration item and ticket.
- How and in which ticket screens the configuration item should be shown (e.g. as a drop-down list or a tree view).
- Visualization in the external or agent interface.

**Benefits**

- Customer-based visualization and selection of configuration items when creating a ticket in the external interface.
- Faster ticket creation thanks to the use of filters to choose which configuration items should be shown, e.g. according to class, usage status, incident status, etc.
- Maximum flexibility through use in different ticket screens for customer users and/or agents.
- Rapid search for configuration items when creating a ticket.
- Useful for statistics and automatic ticket notifications.

**Target Groups**

- Customers of companies that offer services for different devices or products
- Customer service
- Internal and external IT
- Facility management
- Sales
Available in Service Package

- SILVER, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager

- OTRSDynamicFieldCI

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

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

- Ticket Forms

### 29.1 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.

#### 29.1.1 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 Options

Restriction by Interface
Select if the available configuration items should be restricted by customer ID, by customer user ID or by any other attribute in agent and/or external interface.

Configuration Item Field
The name of the configuration item field containing the customer ID, the customer user ID or any other attribute. Usually this is the owner field.

Value Lookup for Customer ID, Customer User ID or Any Attribute
Select if the readable value for the customer ID, the customer user ID or any other attribute field should be used for restriction.

Note: If the filters are changed after the configuration item and the ticket has been linked, the link will not be automatically updated.

Link Options

Link Type
Type of automatically created links for configuration items. None disables the automatic linking.

Link Direction
If a Link Type is selected, choose the link direction from the point of ticket.

Display Options

Multiselect
Activate this option to allow multiple selection of values.

Add Empty Value
Activate this option to create an empty selectable value.

Tree View
Activate this option to display values as a tree.
Fig. 1: Configuration Item-Dynamic Field Settings
Translatable Values
If you activate this option the values will be translated to the user defined language.

Note: You need to add the translations manually into the language translation files.

Autocompletion
Activate this option to enable autocompletion for the values.

Advanced Options

Follow CMDB Links
If you activate this option CMDB items linked to the items set in the Follow Dynamic Field will be displayed as selectable options for this field.

CMDB Link Settings
Define elements to show for this field if no items are selected in the Follow Dynamic Field. Possible values:

- Does not show any items
- Show regular items

Follow Dynamic Field
Dynamic field to follow CMDB links from. This option only works if the setting Restriction by Customer ID or Restriction by Customer User ID is deactivated.

Follow Dynamic Field Attribute
The name of the configuration item field containing the dynamic field value.

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

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

29.2.1 Automatic Linking and Unlinking
When configuration items are added to or removed from tickets via the new dynamic field back end, links for the respective configuration items can be added or removed automatically. If this is configured, links will be compared and links will be added and/or removed, if necessary.

Warning: In order to prevent inconsistencies between the dynamic field content and links please do not add or remove links of the configured type and direction manually via the link mechanism in the ticket detail view. Those changes are not synchronized back to the dynamic field content.

See also:
The dynamic field configuration decides if linking happens and what link type and direction will be used. See the Dynamic Fields in the administrator interface.
29.3 External Interface

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

**Package Name in OTRS Package Manager**

- OTRSDynamicFieldDatabase
30.1 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.

30.1.1 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*.

- **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...
Database Field Settings

Possible values:

- Name:
- Label:
- Datatype:
- Filter:

- Search field
- List field

Add value:

Show link:

Here you can specify an optional HTTP link for the field value displayed in overviews and detail views.
If special characters (&, %, ; /, etc.) should not be encoded, use 'url' instead of 'uri' filter.
Example: http://some.example.com/handle?query=[% Data.Field1 | uri %]

Link for preview:

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

- Type: MySQL
- Server:
- Port:
- Database:
- Table / View:
- User:
- Password:
- Identifier:

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

Multiselect:
Cache TTL:
Search Prefix:
Search Suffix:
Result Limit:
Case Sensitive:
the filters will be ignored. If the filter will be configured to a table column, only search results matching to
the search term and the related ticket attribute on exactly the configured column will be displayed.

**Searchfield**
Indicates if a field should be included in the search requests.

**Listfield**
Indicates if a field should be displayed in the results.

**Add value**
Click on the + icon to add a new section to Possible values.

**Show link**
Here you can specify an optional HTTP link for the field value displayed in overviews and detail views. Examples:

- The value of dynamic field named Field1 is part of the link:
  
  https://some.example.com/handle?query=[% Data.Field1 | uri %]

- The value of dynamic field named Field1 is the complete link with and without protocol:
  
  [% Data.Field1 %]
  https://[% Data.Field1 %]

- The value of dynamic field named 100Field needs special handling, because the name of the dynamic field starts with a number:
  
  https://some.example.com/handle?query=[% Data.item("100Field") | 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 detail view. 
Please note that for this to work, the regular URL field above needs to be filled in, too.

**Type **
The type of the desired database can be selected here. The following 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 (hostname or IP address).

**Port**
The port of the database server.

**Database **
Defines the desired target database of the DBMS. This database will be used for queries.

**Table/View **
This table or view will be used for the queries.

**User **
The username for the database connection.
Password *
The user password for the database connection.

Identifier *
This select box will be automatically filled through Possible values. This field represents the value which will be stored in the dynamic field. Must be unique column from the table entered in Table/View.

Multiselect
If this field is selected, it will be possible to store more than one value to the dynamic field. Those values will be stored comma separated.

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

Search Prefix
This value will be put in the front of every search term while using the auto-completion to search the database. Wildcard characters are supported as well. The search prefix will be ignored during the detailed search, but it is still possible to use wildcard characters in those masks.

Search Suffix
This value will be put in the end of every search term while using the auto-completion to search the database. Wildcard characters are supported as well. The search suffix will be ignored during the detailed search, but it is still possible to use wildcard characters in those masks.

Result Limit
The entered integer value defines the maximum amount of allowed results during a database search. This includes the auto-completion search as well as the detailed search.

Case Sensitive
If this field is selected, case-sensitivity will take effect on searches.

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

Storage of Historical Data

This module offers a functionality to store historical data. For this to work it is necessary to activate and set the settings in the system configuration available in Core → DynamicFieldDatabase navigation path.

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

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

Note: The mapping of multiselect fields to historical data is not supported.
30.2 Agent Interface

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

30.2.1 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 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.

30.3 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

• SILVER, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager

• OTRSDynamicFieldSearchBooster

31.1 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`.

**Note:** This feature is only available to *On-Premise* customers. If you are a *Managed* customer, this feature is taken care of by the *Customer Solutions Team* in OTRS. Please contact us via support@otrs.com or in the OTRS Portal.

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.

### 31.2 Agent Interface

This package has no agent interface.

### 31.3 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

• SILVER, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager

• OTRSDynamicFieldValueImport
32.1 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.

32.1.1 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.

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.

Fig. 1: Value Import Widget
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:

<table>
<thead>
<tr>
<th>Object</th>
<th>Type</th>
<th>Name</th>
<th>Label</th>
<th>Possible values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ticket</td>
<td>Multiselect</td>
<td>Manufacturer</td>
<td>Manufacturer</td>
<td>Leave it empty</td>
</tr>
<tr>
<td>Ticket</td>
<td>Multiselect</td>
<td>Car</td>
<td>Car</td>
<td>Leave it empty</td>
</tr>
<tr>
<td>Ticket</td>
<td>Multiselect</td>
<td>ModelYear</td>
<td>Model Year</td>
<td>Leave it empty</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Manufacturer,Car,ModelYear</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMW,X3,2016</td>
</tr>
<tr>
<td>BMW,318d,2016</td>
</tr>
<tr>
<td>BMW,318d,2017</td>
</tr>
<tr>
<td>BMW,320d,2016</td>
</tr>
<tr>
<td>Audi,A4,2015</td>
</tr>
<tr>
<td>Audi,A4,2016</td>
</tr>
<tr>
<td>Audi,A5,2015</td>
</tr>
<tr>
<td>Audi,A5,2016</td>
</tr>
<tr>
<td>Audi,A6,2016</td>
</tr>
</tbody>
</table>

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 colorized 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.

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.
Fig. 2: Change Overview Widget

Fig. 3: ACL Settings Widget
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.

<table>
<thead>
<tr>
<th>ACL NAME</th>
<th>COMMENT</th>
<th>VALIDITY</th>
<th>EXPORT</th>
<th>COPY</th>
</tr>
</thead>
<tbody>
<tr>
<td>9999-DynamicFieldValueImport-Manufacturer-Car-000</td>
<td>Created automatically by OTRS DynamicFieldValueImport</td>
<td>valid</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9999-DynamicFieldValueImport-Manufacturer-Car-001</td>
<td>Created automatically by OTRS DynamicFieldValueImport</td>
<td>valid</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9999-DynamicFieldValueImport-Manufacturer-Car-002</td>
<td>Created automatically by OTRS DynamicFieldValueImport</td>
<td>valid</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9999-DynamicFieldValueImport-Manufacturer-Car-ModelYear-000</td>
<td>Created automatically by OTRS DynamicFieldValueImport</td>
<td>valid</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9999-DynamicFieldValueImport-Manufacturer-Car-ModelYear-001</td>
<td>Created automatically by OTRS DynamicFieldValueImport</td>
<td>valid</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9999-DynamicFieldValueImport-Manufacturer-Car-ModelYear-002</td>
<td>Created automatically by OTRS DynamicFieldValueImport</td>
<td>valid</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9999-DynamicFieldValueImport-Manufacturer-Car-ModelYear-003</td>
<td>Created automatically by OTRS DynamicFieldValueImport</td>
<td>valid</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9999-DynamicFieldValueImport-Manufacturer-Car-ModelYear-004</td>
<td>Created automatically by OTRS DynamicFieldValueImport</td>
<td>valid</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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).
- Dynamic field values have a length limit of 200 characters per value.
- 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.

### 32.2 Agent Interface

This package has no agent interface.

### 32.3 External Interface

This package has no external interface.
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, GOLD, TITANIUM, PLATINUM

**Package Name in OTRS Package Manager**

- OTRSDynamicFieldWebService

### 33.1 Administrator Interface

This feature implements a generic dynamic field type Web Service that gathers its selectable options from an external system using a web service.

The response from the external system defines the possible options to be displayed, and they could vary depending on the data that is sent in the request. Normally when a field is changed in a screen (e.g. the ticket priority in the New Phone Ticket screen) the values of other fields could be updated. That is the case with this type of dynamic fields, as they could also include all screen field values in the request and the remote server could potentially return completely different values depending on input.
Additionally if the dynamic field source object already exists (e. g. a ticket, and the field is set in the Change Free Fields action), the details of the already created ticket are also included in the request.

33.1.1 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 to 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.

*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.
### Web Service Field Settings

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Web service:</strong></td>
<td>The web service to be executed for possible values.</td>
</tr>
<tr>
<td><strong>Invoker:</strong></td>
<td>The invoker to be used to perform requests (Invoker needs to be of type 'Generic::PassThrough').</td>
</tr>
<tr>
<td><strong>Multiselect:</strong></td>
<td>Activate this option to allow multiselect on results.</td>
</tr>
<tr>
<td><strong>Cache TTL:</strong></td>
<td>Cache time to live (in minutes), to save the retrieved possible values.</td>
</tr>
<tr>
<td><strong>Add empty value:</strong></td>
<td>Activate this option to create an empty selectable value.</td>
</tr>
<tr>
<td><strong>Tree View:</strong></td>
<td>Activate this option to display values as a tree.</td>
</tr>
<tr>
<td><strong>Translatable values:</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Show link:</strong></td>
<td>Here you can specify an optional HTTP link for the field value displayed in overviews and detail views. Optional HTTP link works only for single-select fields. If special characters (&amp;, @, :, /, etc.) should not be encoded, use 'url' instead of 'uri' filter. Example: <a href="http://some.example.com/handle?query=%5B$Data.Field1">http://some.example.com/handle?query=[$Data.Field1</a></td>
</tr>
<tr>
<td><strong>Link for preview:</strong></td>
<td>If filled in, this URL will be used for a preview which is shown when this link is hovered in ticket detail view. Please note that for this to work, the regular URL field above needs to be filled in, too.</td>
</tr>
</tbody>
</table>

---

Fig. 1: Web Service Dynamic Field Settings
Note: You need to add the translations manually into the language translation files.

Show link
Here you can specify an optional HTTP link for the field value displayed in overviews and detail views. Optional HTTP link works only for single-select fields. Examples:

- The value of dynamic field named Field1 is part of the link:

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

- The value of dynamic field named Field1 is the complete link with and without protocol:

  [% Data.Field1 %]
  https://[% Data.Field1 %]

- The value of dynamic field named 100Field needs special handling, because the name of the dynamic field starts with a number:

  [https://some.example.com/handle?query=[% Data.item("100Field") | 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 detail view. Please note that for this to work, the regular URL field above needs to be filled in, too.

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

Web Services
To create a new dynamic field of type Web Service it is necessary to have an already working web service. It requires to have at least one invoker of the type Generic::PassThrough. This invoker will be called to fetch the data from the remote server. The original data that it is sent in a request is similar to the following example.

```json
{
  DynamicFieldID => 123,
  DynamicFieldLabel => 'NameX',
  DynamicFieldName => 'NameX',
  DynamicFieldValue => 'Value',
  Form => {
    # Form fields
    # ...
  },
  Ticket => {
    # Ticket attributes
    # ...
  },
  DynamicField => {
    NameX => 'Value'
    NameY => ['Value']
  },
  UserID => 123,
}
```
Form
This section contains the fields in the current form in the web browser. This information changes as the screen is filled in.

Ticket
This section (or another source object, e.g. CustomerUser) contains the attributes of the object where the dynamic field belongs.

For example in the New Phone Ticket screen the section is empty as the ticket is not created yet, but in the Change Free Fields screen it contains the information of the current ticket.

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

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

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

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:transform
    xmlns:xsl="https://www.w3.org/1999/XSL/Transform"
    xmlns:date="https://exsalt.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>somепassword</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://exsalt.org/dates-and-times"
    version="1.0"
    extension-element-prefixes="date">

</xsl:transform>
```

(continues on next page)
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.
Transport:
  Config:
    AdditionalHeaders: ~
    MaxLength: '100000000'
    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) 2001-2023 OTRS AG, https://otrs.com/
      This software comes with ABSOLUTELY NO WARRANTY. For details, see
      the enclosed file COPYING for license information (GPL). If you
      did not receive this file, see https://www.gnu.org/licenses/gpl.txt.
      -->

      <!-- DOCUMENTATION

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

      * Example XML Output *
      <RootElement>
        <PossibleValues>
          <Key>???</Key>
          <Value>???</Value>
        </PossibleValues>
        <PossibleValues>
          <Key>???</Key>
          <Value>???</Value>
        </PossibleValues>
        ...
      </RootElement>

      -->

      <xsl:transform
        xmlns:xsl="https://www.w3.org/1999/XSL/Transform"
        xmlns:date="https://exslt.org/dates-and-times"
        version="1.0"
<Key>???</Key>
<Value>???</Value>
</PossibleValues>
...
</RootElement>

-->

<xsl:transform
xmlns:xsl="https://www.w3.org/1999/XSL/Transform"
xmlns:date="https://exslt.org/dates-and-times"
version="1.0"
extension-element-prefixes="date">
<xsl:output method="xml" encoding="utf-8" indent="yes" />

<!-- Don't return unmatched tags -->
<xsl:template match="text()" />

<!-- Remove empty elements -->
<xsl:template match="*[not(node())]" />

<!-- Root template -->
<xsl:template match="/">

<RootElement>
<UserLogin>someuser</UserLogin>
>Password>sometpassword</Password>
<TicketID>1</TicketID>

</RootElement>
</xsl:template>

</xsl:transform>

**Type:** XSLT

**Type:** Generic::PassThrough

**Transport:**

**Config:**

**Encoding:** ''

**Endpoint:** https://localhost/otrs/nph-genericinterface.pl/Webservice/

**GenericConfigItemConnectorSOAP**

**NameSpace:** https://www.otrs.org/TicketConnector/

**RequestNameFreeText:** ''

**RequestNameScheme:** Plain

**ResponseNameFreeText:** ''

**ResponseNameScheme:** Response

**SOAPAction:** Yes

**SOAPActionSeparator:** '#'

**SSL:**

**SSLProxy:** ''

**SSLProxyPassword:** ''

**SSLProxyUser:** ''

**Type:** HTTP::SOAP

**UseMappedData:** '1'
33.2 Agent Interface

This package has no agent interface.

33.3 External Interface

This package has no external interface.
CHAPTER
THIRTYFOUR

DYNAMIC POPUP WINDOW SIZE

Warning: This feature add-on has reached the end of life with OTRS 7 and will not be longer supported.

The same functionality can be easily realized using the actions and popup windows of the new agent 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

• SILVER, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager

• OTRSDynamicSenderAddresses
### 35.1 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.

#### 35.1.1 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.

![Sender Address Mapping Management Screen](image)

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:

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

### 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.

**System address** *

Select a system email address configured in Email Addresses module in the administrator interface.

### 35.1.2 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: ServiceCopy
   - 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 key: Ticket_Service

---

Fig. 5: DynamicSenderAddresses###Mapping Setting
– Param value: ServiceCopy

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.

### 35.2 Agent Interface

This package has no agent interface.

#### 35.2.1 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:** Copy service value to dynamic field value

**Validity:** Yes

#### Automatic Execution (Multiple Tickets)

#### Event Based Execution (Single Ticket)

<table>
<thead>
<tr>
<th>TYPE</th>
<th>EVENT</th>
<th>DELETE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ticket</td>
<td>TicketCreate</td>
<td></td>
</tr>
<tr>
<td>Ticket</td>
<td>TicketServiceUpdate</td>
<td></td>
</tr>
</tbody>
</table>

Additionally or alternatively to a periodic execution, you can define ticket events that will trigger this job. If a ticket event is fired, the ticket filter will be applied to check if the ticket matches. Only then the job is run on that ticket.

**Add Event Trigger:**

- Ticket
- TicketServiceUpdate

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

<table>
<thead>
<tr>
<th>Module:</th>
<th>Kernel::System::GenericAgent::CopyTicketValueToDynamicField</th>
</tr>
</thead>
<tbody>
<tr>
<td>Param 1 key:</td>
<td>Ticket_Service</td>
</tr>
<tr>
<td>Param 2 key:</td>
<td></td>
</tr>
<tr>
<td>Param 3 key:</td>
<td></td>
</tr>
<tr>
<td>Param 4 key:</td>
<td></td>
</tr>
<tr>
<td>Param 5 key:</td>
<td></td>
</tr>
<tr>
<td>Param 6 key:</td>
<td></td>
</tr>
</tbody>
</table>

### Save Changes

- Save
- Cancel

---

Fig. 6: Generic Agent Configuration Example

Chapter 35. Dynamic Sender Addresses
35.2.2 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).

35.3 External Interface

This package has no external interface.
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**

- SILVER, GOLD, TITANIUM, PLATINUM

**Package Name in OTRS Package Manager**

- OTRSEscalationSuspend

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

- Advanced Escalations
36.1 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).

36.1.1 Configuration

To configure the states where the escalation is stopped:

1. Go to the 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.

36.1.2 Console Command

Note: This feature is only available to On-Premise customers. If you are a Managed customer, this feature is taken care of by the Customer Solutions Team in OTRS. Please contact us via support@otrs.com or in the OTRS Portal.

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.

```bash
otrs> /opt/otrs/bin/otrs.Console.pl Maint::Daemon::Summary
```

36.2 Agent Interface

This package has no dedicated agent interface.

Note: For this example escalations should be configured for at least one queue.

Example usage:
OTRS Feature Add-On Manual, Release 2023.1.1

1. Go to one of the new ticket screens, and create a new ticket for one of the queues configured with escalation (for this example *Escalation - first response time* is OK).

2. Open the ticket detail view for the newly created ticket. Notice that the escalation is running.

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 detail view 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.

There is a new filter *Suspended Escalations* for the ticket lists. Select the *Remove tickets with suspended escalations* checkbox to remove all tickets from the list that are in an escalation suspending state.

---

**Fig. 1: Suspended Escalations Filter in Ticket List**

Several ticket list screens also provide a default filter preset called *Escalation is not suspended*, which applies the aforementioned filter.

---

**Fig. 2: Filter Preset in *My Escalations* Organizer Item**

A suspended escalation can be seen in the ticket list, if hovering on an entry within the column *Escalation Time*. In the ticket detail view of a suspended ticket, the suspension status is visible in a property card.
36.3 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

• SILVER, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager

• OTRSExtendedTicketStats
37.1 Administrator Interface

This package has no administrator interface.

37.2 Agent Interface

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

37.2.1 Statistics and Reports

After installation of the package, a new statistic and two new list objects will be available for the statistics.

New Statistic

After installation of the package, a new static statistic will be available in the Static widget of the Statistics and Reports screen.

![Static Widget](image)

Fig. 1: Static Widget

Response Statistic

This static type statistic makes it possible to count how many times a response template has been used on a time frame. The result can be filtered by a time period and by queues.

To use the new statistic:

1. Search for Response Statistic in the Static widget.
2. Click on the Run icon in the column.
3. Select the time frame and the queues to include in the statistic.
4. Generate your statistic.

New List Objects

After installation of the package, two new list objects will be added to the system.
Configurable parameters of static statistic

Start Day
01

Start Month
03

Start Year
2020

End Day
01

End Month
04

End Year
2020

Queues
Junk
Misc

Fig. 2: Response Statistic Settings
Statistics for Queue Resting Time

You are able to calculate the resting time of tickets for queues with this statistic module. The statistic considering also the close times of tickets. This accounted time will not added to the resting time or the resting time based on working period. The resting time based on working periods will be only displayed if you have configured calendars for your queues.

Available columns are:

- Number: numeration of the ticket list.
- Ticket#: ticket number.
- Ticket ID: internal ID of the ticket.
- Created: ticket creation date.
- Changed: last change date of the ticket.
- Closed: last close date of the ticket.
- Status: current state of the ticket.
- Type: current type of the ticket.
- Queue: current queue based on the history entries.
- Queue join time: date based on history entries (ticket created in this queue, moved into this queue while being open or reopened in this queue).
- Queue left time: date based on history entries (ticket closed in this queue or moved out of this queue while being open). Empty if ticket was still open and in this queue at end of reporting time-frame.
- Resting time (seconds): count of the duration when the ticket stayed in the queue, based on join and left time. Times in which a ticket was closed will not be counted.
- Resting time (HH:MM:SS): count of the duration when the ticket stayed in the queue, formatted as hours, minutes and seconds.
- Resting time (Working hours, seconds): count of the duration when the ticket stayed in the queue based on the queue configured working calendars in the system configuration (this field will be empty if no calendar is configured).
- Resting time (Working hours, HH:MM:SS): formatted count of the duration when the ticket stayed in the queue based on the queue configured working calendars in the system configuration (this field will be empty if no calendar is configured).
- Criticality: current criticality of the ticket.
- Impact: current impact of the ticket.

To use the new list object:

1. Create a new List statistic.
2. Choose the object OTRSStatsQueueRestingTime.
3. Configure the rest of the configuration options according to your wishes.
4. Run the statistic with output print or CSV.
5. All tickets matching the restrictions will be shown. Additional tickets will be shown, if they were created before the beginning of the reporting and were still open during the reporting time frame.
Statistics for Queue Working Time

This statistic is a modified ticket list to calculate the working time per queue. This modified ticket list statistic contains 3 new columns for the X-axis. In general the time for the 3 new columns is calculated on the time where ticket was not in a closed, removed or merged state type.

All queues in the system have the Time, Working Time and Effective Time column for this statistic object.

Example queue Misc:

- Queue Misc Time contains the time the agent have worked on the ticket.
- Queue Misc Working Time contains the time the agent have worked on the ticket based on the working calendars of the queue Misc.
- Queue Misc Effective Time contains the time the agent set as time units for the ticket while it was in the queue Misc.

The values are displayed for each agent by first and last name in the same column (comma separated).

To use the new list object:

1. Create a new List statistic.
2. Choose the object OTRSStatsQueueWorkingTime.
3. In the X-axis section, select the columns for the queue you like to display the working time or time.
4. Set all other restrictions and sort options you need for the statistic to complete the configuration.
5. Generate your statistic.

37.3 External Interface

This package has no external interface.
This feature add-on supports agents in keeping track of all incidents and requests that relate to a specific focus topic. This is especially helpful when many tickets are handled by different agents at the same time.

With this feature add-on, agents can easily and quickly identify all relevant tickets by defining the ticket properties in their personal settings, that belong to their focus topics. For each selected focus topic, agents can receive notifications, for example, when tickets are created or SLA times are reached. The available options for defining focus topics are not limited to services but include all ticket properties as well as dynamic fields. The ticket properties for the focus topic settings in the agent preferences are enabled/disabled in the system configuration.

**Benefits**
- Better overview of incidents and requests.
- Better task organization.
- Focused notifications.

**Target Groups**
- Large companies
- Departments with high ticket volume
- Departments with many employees
- Service-oriented workgroups
- Companies with outsourced first level support

**Available in Service Package**
- SILVER, GOLD, TITANIUM, PLATINUM

**Package Name in OTRS Package Manager**
- OTRSFocusTopics

### 38.1 Administrator Interface

This package has no administrator interface, but adds some new entries to the recipient list of ticket notifications.
38.1.1 Communication & Notifications

This feature add-on adds new notification recipients for agents, who set their focus topics in the *Personalization*.

**Note:** To be able to use all options in focus topic settings, you need to enable the following system configuration options:

- Ticket::Responsible
- Ticket::Service
- Ticket::Type

**Ticket Notifications**

This feature add-on adds new notification options for the agents in the *Recipients* section.

**Ticket Notification Settings**

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

![Fig. 1: Ticket Notification Settings - Recipients](image)

**Send to**

The focus topics of all assigned agents can be chosen with the following options to notify the agents:

- All agents who focus on the owner
- All agents who focus on the priority
- All agents who focus on the queue
- All agents who focus on the responsible
- All agents who focus on the service
- All agents who focus on the SLA
- All agents who focus on the state
- All agents who focus on the type
- All agents who focus on the value
All other setting and options are part of the OTRS framework.

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

### 38.2 Agent Interface

This feature allows agents to set various ticket attributes as focus topics. Each agent can focus on multiple ticket attributes and each ticket attribute can be focused on by multiple agents. Some new filter options are available for ticket lists to filter for focus topics.

#### 38.2.1 Personalization

With this feature add-on each agent can select various ticket attributes within the related user preferences. The configuration screen is available in the *Focus Topic Settings* menu item of the *Personalization* menu.

![Personalization](image)

**Focus Topic Settings**

These settings are for customizing personal notifications. In this screen the agents can set various ticket attributes to focus on those tickets.

Select one or more ticket attributes from the drop-down lists.

**Note:** *Ticket Notifications* have to be also configured by the system administrator in order to receive the notifications.
Focus Topic Settings

Start typing to filter...

Queue

Select...

Queues are like folders for tickets. A ticket can belong to only one queue at a time.

Type

Select...

Service

Select...

SLA

Select...

Fig. 3: Focus Topic Settings
38.2.2 Navigation And Usage

This feature add-on adds new filter options for ticket lists.

The following new filter options are available:

- Priorities I focus on
- Queues I focus on
- Responsibles I focus on
- SLAs I focus on
- Services I focus on
- States I focus on
- Types I focus on
- Values I focus on

38.3 External Interface

This package has no external interface.
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.

With this feature add-on you can build flexible and dynamic forms by hiding or showing fields based on ticket information, permissions and more. This is controlled by ticket access control lists.

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**

- SILVER, GOLD, TITANIUM, PLATINUM

**Package Name in OTRS Package Manager**

- OTRSHideShowDynamicFields

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

- Ticket Forms
39.1 Administrator Interface

This package has no administrator interface, but the function can be used in agent and external interface. The following screens can be configured.

Agent interface:

- New Email Ticket
- New Phone Ticket
- New Process Ticket (incl. Process Activity Dialogs)
- New SMS Ticket
- Close Ticket
- Change Customer
- Send Email Outbound
- Change Free Fields
- Link Objects
- Merge Ticket
- Move Ticket
- Add Note
- Change Owner
- Set Pending Time
- Add Phone Call Inbound
- Add Phone Call Outbound
- Change Priority
- Change Responsible
- Send SMS Outbound
- Forward via Email
- Redirect via Email
- Reply via Email
- Reply to All via Email
- Reply via Note
- Reply via SMS

External interface:

- Create New Ticket
- Create New Process Ticket (incl. Process Activity Dialogs)
- Ticket Reply (within Ticket Detail View)

**Note:** This feature works with ticket dynamic fields only. Other dynamic field object types are not supported.
39.1.1 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:
<table>
<thead>
<tr>
<th>Object</th>
<th>Type</th>
<th>Name</th>
<th>Label</th>
<th>Possible values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ticket</td>
<td>Dropdown</td>
<td>Brand</td>
<td>Brand</td>
<td>• VW → VW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Peugeot → Peugeot</td>
</tr>
<tr>
<td>Ticket</td>
<td>Dropdown</td>
<td>VWModel</td>
<td>VW Model</td>
<td>• Up → Up</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Polo → Polo</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Golf → Golf</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• T5 → T5</td>
</tr>
<tr>
<td>Ticket</td>
<td>Dropdown</td>
<td>VWProductionFacility</td>
<td>VW Production Facility</td>
<td>• Barcelona → Barcelona</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Berlin → Berlin</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Bratislava → Bratislava</td>
</tr>
<tr>
<td>Ticket</td>
<td>Dropdown</td>
<td>PeugeotModel</td>
<td>Peugeot Model</td>
<td>• 207 → 207</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• 307 → 307</td>
</tr>
<tr>
<td>Ticket</td>
<td>Dropdown</td>
<td>PeugeotProductionFacility</td>
<td>Peugeot Production Facility</td>
<td>• Poissy → Poissy</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Madrid → Madrid</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Trnava → Trnava</td>
</tr>
<tr>
<td>Ticket</td>
<td>Dropdown</td>
<td>Fuel</td>
<td>Fuel</td>
<td>• Gasoline → Gasoline</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Diesel → Diesel</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Gas → Gas</td>
</tr>
<tr>
<td>Ticket</td>
<td>Multiselect</td>
<td>Accessories</td>
<td>Accessories</td>
<td>• CDRadio → CD Radio</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• GPS → GPS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• ProximitySensors → Proximity Sensors</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• RearCamera → Rear Camera</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• ClimateControl → Climate Control</td>
</tr>
<tr>
<td>Ticket</td>
<td>Textarea</td>
<td>Remarks</td>
<td>Remarks</td>
<td></td>
</tr>
<tr>
<td>Ticket</td>
<td>Date</td>
<td>RegistrationDate</td>
<td>Registration Date</td>
<td></td>
</tr>
<tr>
<td>Ticket</td>
<td>Date</td>
<td>InvoiceDate</td>
<td>Invoice Date</td>
<td></td>
</tr>
</tbody>
</table>
Note: All Dropdown and Multiselect dynamic fields should have the option Add empty value set to Yes in their configuration.

Add the dynamic fields to the New Phone Ticket screen via setting `Forms###AgentFrontend::TicketCreate::Phone::CreateProperties`:

- **Label:** Dynamic Fields
- **Collapsible:** 1
- **Fields:**
  - **Name:** DynamicField_Brand
  - **Name:** DynamicField_VWModel
  - **Name:** DynamicField_VWProductionFacility
  - **Name:** DynamicField_PeugeotModel
  - **Name:** DynamicField_PeugeotProductionFacility
  - **Name:** DynamicField_Fuel
  - **Name:** DynamicField_Accessories
  - **Name:** DynamicField_Remarks
  - **Name:** DynamicField_RegistrationDate
  - **Name:** DynamicField_InvoiceDate

Import this ACL:

```
---
- ChangeBy: root@localhost
  Comment: '
  ConfigChange:
    PossibleNot:
      Form:
        - PeugeotModel
        - PeugeotProductionFacility
        - Accessories
        - Fuel
        - Remarks
        - RegistrationDate
        - InvoiceDate
    ConfigMatch:
      Properties:
        Ticket:
          DynamicField_Brand:
          - VW
  CreateBy: root@localhost
  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**.

39.1. Administrator Interface
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:

---
- CreateBy: root@localhost
- CreateTime: 2019-07-22 11:47:02
- Description: ''
- ID: 2
- Name: ACL-VW-Up

(continues on next page)
Detailed explanation:

**DynamicField_Brand:**
- VW

**DynamicField_VWModel:**
- Up

In this example two conditions should be met. Brand has to be **VW** and VW model has to be **Up** for this rule to come into action. It will be triggered only if an agent selects brand **VW** and **VW model Up**.

**PossibleAdd:**
- Form:
  - Accessories
  - Fuel
  - Remarks
  - RegistrationDate
  - InvoiceDate

Here the dynamic fields **VW Model** and **VW Production Facility** were already visible and they remain, but **Accessories, Fuel, Remarks, Registration Date and Invoice Date** has to be re-added to the fields that are visible. This is done in the **PossibleAdd** section as the first ACL sets this fields as not shown and both ACLs works together. If this was done in the **Possible** section for example, the result will be that only these fields explicitly will be shown and **VW Model** and **VW Production Facility** will be hidden as they are not longer part of the (new) **Possible** section.

**PossibleNot:**
- Form:
  - PeugeotModel
  - PeugeotProductionFacility

Just **Peugeot Model** and **Peugeot Production Facility** are invisible (in our example it does not make much sense to configure a Peugeot model if the user has a VW Up).

In addition to the visibility of dynamic fields there is the possibility to show just some of the possible values of a dynamic field. Combined into ACL rules like in here, makes it easier to handle big multiselects.

**Possible:**
- Ticket:
  - DynamicField_Accessories:
    - CDRadio
    - ClimateControl
  - DynamicField_Fuel:
    - Gasoline
  - 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.

### 39.2 Agent Interface

This package has no agent interface.

### 39.3 External Interface

This package has no external interface.
LDAP PASSWORD NOTIFICATIONS

This package displays the directory service password policy password expiration message that was previously displayed to users only after they logged on to corporate workstations.

Because of the messages, users can keep their passwords up to date so they can avoid having to update them after log in. This is especially important for users, either agent or a customer user, who need to respond to a ticket in a very short time. In this case, the user can always respond quickly to messages because they do not have to spend valuable minutes updating their expired password.

Notifications can be displayed in several phases, which can be individually configured in the system settings. In the first phase, OTRS informs users via a warning message a few days before the password expires, telling them how many days are left until it needs to get changed.

If companies provide a separate website for updating passwords, it is possible to configure notifications with a link to that website so that users can update their expiring passwords for the notification immediately.

**Benefits**

- Agents and customer users are informed about the required password change.
- No time wasted due to expired passwords.
- Always a quick response.

**Target Groups**

- External IT
- Internal IT
- Any company uses directory service
- 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**

- SILVER, GOLD, TITANIUM, PLATINUM

**Package Name in OTRS Package Manager**

- OTRSLDAPPasswordNotifications
40.1 Administrator Interface

This package has no administrator interface.

40.2 Agent Interface

This package has no agent interface.

A notification is displayed when the LDAP password is about to expire or the LDAP password is expired.

Fig. 1: Notification on Agent Interface

40.3 External Interface

This package has no external interface.

A notification is displayed when the LDAP password is about to expire or the LDAP password is expired.

Fig. 2: Notification on External Interface
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**
- SILVER, GOLD, TITANIUM, PLATINUM

**Package Name in OTRS Package Manager**
- OTRSLinkObjectConnector

### 41.1 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.

### 41.1.1 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 built-in 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
Link::PossibleObjectsList
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.
41.1.2 Usage

The link object feature provides functionality to link items (e.g. tickets, changes) from objects (e.g. Ticket, ITSM-ChangeManagement) 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>
</RootElement>
```

(continues on next page)
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

<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

<Object2>?</Object2>
<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.

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

<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

<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.

41.1.3 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.

```
<namesp1:RootElementResponse>
  <Success>1</Success>
</namesp1:RootElementResponse>
```
Remove Link

This example will show how to remove the previously created link, using the \texttt{Link::LinkDelete} generic interface operation.

1. Send the following SOAP request to the OTRS system.

   \begin{verbatim}
   <RootElement>
     <UserLogin>johnsmith</UserLogin>
     <Password>Secret123</Password>
     <Object1>Ticket</Object1>
     <Key1>1</Key1>
     <Object2>Ticket</Object2>
     <Key2>2</Key2>
     <Type>Normal</Type>
   </RootElement>
   \end{verbatim}

2. The server response should be something similar to this one.

   \begin{verbatim}
   <namesp1:RootElementResponse>
     <Success>1</Success>
   </namesp1:RootElementResponse>
   \end{verbatim}

Remove All Links

This example will show how to alternatively remove the previously created link (and all other links of this ticket), using the \texttt{Link::LinkDeleteAll} generic interface operation.

1. Send the following SOAP request to the OTRS system.

   \begin{verbatim}
   <RootElement>
     <UserLogin>johnsmith</UserLogin>
     <Password>Secret123</Password>
     <Object>Ticket</Object>
     <Key>1</Key>
   </RootElement>
   \end{verbatim}

2. The server response should be something similar to this one.

   \begin{verbatim}
   <namesp1:RootElementResponse>
     <Success>1</Success>
   </namesp1:RootElementResponse>
   \end{verbatim}

Show All Links

This example will show all links of an object, using the \texttt{Link::LinkList} generic interface operation.

1. Send the following SOAP request to the OTRS system.

   \begin{verbatim}
   <RootElement>
     <UserLogin>johnsmith</UserLogin>
     <Password>Secret123</Password>
     <Object>Ticket</Object>
     <Key>1</Key>
     <State>Valid</State>
   </RootElement>
   \end{verbatim}
2. The server response should be something similar to this one.

```xml
<namesp1: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>
  </RootElement>
</namesp1: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.

```xml
<RootElement>
  <UserLogin>johnsmith</UserLogin>
  <Password>Secret123</Password>
</RootElement>
```

2. The server response should be something similar to this one.

```xml
<namesp1:RootElementResponse>
  <RootElement>
    <Object1>Ticket</Object1>
    <Object2>Ticket</Object2>
    <Type>Normal</Type>
  </RootElement>
  <RootElement>
    <Object1>Ticket</Object1>
    <Object2>Ticket</Object2>
    <Type>ParentChild</Type>
  </RootElement>
</namesp1: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.

```xml
<RootElement>
  <UserLogin>johnsmith</UserLogin>
  <Password>Secret123</Password>
  <Object>Ticket</Object>
</RootElement>
```

2. The server response should be something similar to this one.

```xml
<namesp1:RootElementResponse>
  <PossibleObject>Ticket</PossibleObject>
</namesp1: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.

```xml
<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.

```xml
<namesp1:RootElementResponse>
  <PossibleType>Normal</PossibleType>
  <PossibleType>ParentChild</PossibleType>
</namesp1:RootElementResponse>
```

41.2 Agent Interface

This package has no agent interface.
41.3 External Interface

This package has no external interface.
MOBYDICK CONNECTOR

**Warning:** This feature add-on has reached the end of life with OTRS 7 and will not be longer supported.
OUT OF OFFICE

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**

- SILVER, GOLD, TITANIUM, PLATINUM

**Package Name in OTRS Package Manager**

- OTRSOutOfOffice

### 43.1 Administrator Interface

This package has no administrator interface.
43.1.1 Configuration

This package includes the functionality to create articles for a ticket with the X-OTRS-OutOfOffice header 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 the System Configuration screen.
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-OutOfOffice 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.

43.2 Agent Interface

This package has no agent interface.

43.3 External Interface

This package has no external interface.
This package adds support to define a primary ticket and some secondary tickets that follow their primary ticket updates:

- Replies (sent new articles as email)
- Notes (any new article)
- State changes (including pending time set)
- Priority changes
- Owner changes
- Responsible changes
- Lock status changes

The primary/secondary relations are very flexible. It is possible to create new primary ticket, remove the primary or the secondary status or assign a ticket as the secondary ticket of another ticket.

**Note:** This package has been renamed in this release. The former name was Master / Slave until OTRS 7.

**Available in Service Package**

- SILVER, GOLD, TITANIUM, PLATINUM

**Package Name in OTRS Package Manager**

- OTRSPrimarySecondary

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

- Ticket Watchlist

### 44.1 Administrator Interface

This package has no administrator interface.
44.1.1 New Dynamic Field

After installation of the package a new dynamic field of type *Primary / Secondary* will be added to the system where the primary/secondary information is stored. The creation of more fields of this type is not possible.

The *Primary / Secondary* dynamic field is added to the ticket create screens. This makes it possible to set the primary or the secondary status of a ticket at ticket creation time.

To remove the *Primary / Secondary* dynamic field from the ticket create screens:

1. Go to the *System Configuration* screen.
2. Search for the setting `Forms###AgentFrontend::TicketCreate::Phone::CreateProperties`.
3. Change the related form setting to remove the dynamic field from the screen. Since this dynamic field is a special dynamic field, the name is not `DynamicField_PrimarySecondary`, but it is just `PrimarySecondary`.

   Example:

   ```
   - Name: PrimarySecondary
   ```

Same applies for the process management. The field which needs to be added to a user task activity dialog is called `PrimarySecondary`, not `DynamicField_PrimarySecondary`.

44.1.2 Exception for Closing Parent Ticket

The `Ticket::Acl::Module###1-Ticket::Acl::Module` is not compatible to use with this package by default. To allow closing parent tickets only if all its children are already closed, an additional ACL have to be added which makes an exception for primary/secondary tickets.

Here is an example ACL for this exception:

```---
- ChangeBy: root@localhost
  ChangeTime: 2021-09-07 13:45:58
  Comment: Exception for primary/secondary tickets.
  ConfigChange:
    PossibleAdd:
      Endpoint:
        - AgentFrontend::Ticket::Action::Close
      Ticket:
        State:
          - '[regexp]close'
  ConfigMatch:
    PropertiesDatabase:
      DynamicField:
        DynamicField_PrimarySecondary:
          - Primary
  CreateBy: root@localhost
  CreateTime: 2019-05-24 11:45:29
  Description: 'This ACL allows using the Primary/Secondary feature together with the "Ticket::Acl::Module###1-Ticket::Acl::Module" setting.'
  ID: 1
  Name: Primary/Secondary Exception
  StopAfterMatch: 0
  ValidID: 1
```
44.1.3 ACL Reference

This package adds new endpoints that can be used in ACLs. The following reference lists all endpoints added by this package and extends the existing ACL reference from the Administrator Manual with the new entries only.

---

**ChangeBy:** root@localhost

**ChangeTime:** 2021-09-13 13:51:29

**Comment:** ACL Reference for Primary/Secondary.

**ConfigMatch:**

**Properties:**

**DynamicField:**

DynamicField_PrimarySecondary:
- Primary
- SecondaryOf:TICKET_NUMBER (where TICKET_NUMBER is a primary ticket number)

**Frontend:**

**Endpoint:**

- AgentFrontend::Ticket::Action::AddSecondaries
- AgentFrontend::Ticket::Action::EmailToSecondaries
- AgentFrontend::Ticket::Action::MoveToPrimary
- AgentFrontend::Ticket::Action::NoteToSecondaries
- AgentFrontend::Ticket::Action::SetOnlyToPrimary
- AgentFrontend::Ticket::Action::SetToPrimary
- AgentFrontend::Ticket::Action::SetToSecondary
- AgentFrontend::Ticket::Action::SmsToSecondaries
- AgentFrontend::Ticket::Action::UnsetPrimary
- AgentFrontend::Ticket::Action::UnsetSecondary
- AgentFrontend::Ticket::Action::UpdateSecondaries

**PropertiesDatabase:**

# Match properties (existing values from the database).
# Please note that Frontend is not in the database, but in the framework.
# See section "Properties", the same configuration can be used here.

**ConfigChange:**

**Possible:**

# Reset possible options (white list).

**Endpoint:**

# Limit the functions on agent interface.

- AgentFrontend::Ticket::Action::AddSecondaries
- AgentFrontend::Ticket::Action::EmailToSecondaries
- AgentFrontend::Ticket::Action::MoveToPrimary
- AgentFrontend::Ticket::Action::NoteToSecondaries
- AgentFrontend::Ticket::Action::SetOnlyToPrimary
- AgentFrontend::Ticket::Action::SetToPrimary
- AgentFrontend::Ticket::Action::SetToSecondary
- AgentFrontend::Ticket::Action::SmsToSecondaries
- AgentFrontend::Ticket::Action::UnsetPrimary
- AgentFrontend::Ticket::Action::UnsetSecondary
- AgentFrontend::Ticket::Action::UpdateSecondaries

**Ticket:**

# Possible ticket options (white list).

DynamicField_PrimarySecondary:
- Primary
- SecondaryOf:TICKET_NUMBER (where TICKET_NUMBER is a primary ticket number)

**PossibleAdd:**

# Add options (white list).

# See section "Possible", the same configuration can be used here.

**PossibleNot:**

# Remove options (black list).

---

(continues on next page)
44.2 Agent Interface

This package adds new actions to the ticket detail view to set and change the primary/secondary status and to update the secondary tickets. Bulk actions are available to do the same in ticket lists.

Note: The ticket handling with Primary / Secondary has significantly changed in comparison to Master / Slave. The major changes are:

- Changed: Normal ticket updates will not be applied to secondary tickets. The agent have to use the primary/secondary ticket actions to replicate the actions done in a primary ticket.
- New: Beside replications now the agent can add information and/or messages to the secondary tickets only without adding the information or messages to the primary ticket.
- New: Added bulk actions.
- New: Filter for primary/secondary tickets in any ticket list.

44.2.1 Create Primary / Secondary Relation

To create a primary ticket:

1. Go to the ticket detail view.
2. Select the Set to Primary or the Set Only to Primary action in the Actions menu.
3. Click on the Set button.

The Set to Primary action makes it possible to set secondary tickets at the same time as well as to add a message if needed. If the Properties section is configured, additional information can be added.

Using the Set Only to Primary action sets the ticket as primary immediately without confirmation. In this case, secondary tickets can be added later.

To create a secondary ticket:

1. Go to the ticket detail view.
2. Select the Set to Secondary action in the Actions menu.
3. Select a primary ticket from the list and add a message if needed.
4. Click on the Set button.

It is possible to move a primary ticket as secondary ticket to a different primary ticket. In this case the old secondary tickets of the primary ticket will be unset.

To move a secondary ticket to a different primary ticket:
1. Go to the ticket detail view of the secondary ticket.
2. Select the *Move to Different Primary* action in the *Actions* menu.
3. Select another primary ticket and add a message if needed.
4. Click on the *Move* button.

To unset a primary ticket:
1. Go to the ticket detail view of the primary ticket.
2. Select the *Unset Primary* action in the *Actions* menu.
3. Click on the *Unset* button.

With this ticket action the primary ticket can be set to a normal ticket which is no primary or secondary any more.

With the checkbox *Keep secondary tickets as linked tickets* it can be controlled that the link between the unset primary ticket and its secondary ticket shall be kept or not.

If message has been added or if some properties are changed via the configured *Properties* section it can be controlled whether the message shall be replicated to the old primary ticket and/or to the secondary tickets. The first behavior can be set with the checkbox *Update old primary* while the latter can be set with the checkbox *Update secondaries*.

There is a *Replace Primary and Move Secondaries* button which makes possible to move the secondary tickets to an other primary ticket. Without this, the secondary status of the secondary tickets will be also unset.

To unset a secondary ticket:
1. Go to the ticket detail view of the secondary ticket.
2. Select the *Unset Secondary* action in the *Actions* menu.
3. Click on the *Unset* button.

If the *Keep primary ticket as linked ticket* checkbox is checked, only the secondary status of the ticket will be removed and the existing link to the primary ticket will be kept.

The primary/secondary relations are based on *Primary / Secondary* dynamic field using the *ParentChild* link type.

### 44.2.2 Update or Replicate Updates to Secondaries

Normal ticket updates will not be applied to secondary tickets. The agent have to use the primary/secondary ticket actions to replicate the actions done in a primary ticket.

For updating secondary tickets or replicating information from the primary ticket to the secondary tickets the following ticket actions are available:

- Update Secondaries
- Add Note to Secondaries
- Send Email to Secondaries
- Send SMS to Secondaries (if SMS is available in the system)

To update secondary tickets from a primary ticket:
1. Go to the ticket detail view of the primary ticket.
2. Select the *Update Secondaries* action in the *Actions* menu.
3. Change the priority to *1 very low* and add a message.
4. Check the *Update primary ticket* checkbox.
5. Go to the ticket detail view of the secondary ticket.

6. The priority of the secondary ticket has been also changed to *very low* and the same article has been added to this ticket as the primary ticket has.

For this example the priority of the primary ticket has been changed, but there are other activities that can be done instead to test this behavior, like change the state, the owner, the responsible, the lock status, reply the ticket, add a note or close the ticket.

When a new article is created on the primary ticket, the secondary tickets replicate the article using the own customer user.

When updating the secondary tickets, the agents can decide if they want to update the primary ticket as well. There is a checkbox in each update action of the secondary ticket to control this behavior.

### 44.2.3 Primary / Secondary Bulk Actions

A new primary/secondary icon has been added to the breadcrumb bar of each ticket lists to enable the bulk action. The bulk action screen enables the following actions:

**Add to New Primary**
- To add the selected tickets to a newly created primary ticket.

**Unset Primary**
- To unset the current primary ticket.

**Set to Secondary**
- To assign the selected tickets as secondary ticket of a primary ticket.

**Unset Secondary**
- To unset secondary tickets.

Using this bulk action screen is similar to other bulk actions. The agents have to have proper permission to tickets that are affected by the bulk action.

There are two options that may appear while using bulk actions. Displaying the options depends on the chosen primary/secondary bulk action and on the primary/secondary status.

**Move secondary tickets to new primary**
- If this option is checked, the secondary tickets of the old primary ticket will be moved to the new primary ticket. Without this, the secondary status of the secondary tickets will be unset.

**Keep secondary tickets as linked tickets**
- If this option is checked, the existing links between the tickets will be kept.

### 44.2.4 Ticket Create Screens

If the *Primary / Secondary* dynamic field is added to the ticket create screen, the agent can set the primary/secondary status of the new ticket at creation time.
44.2.5 Dashboard Widget

This package adds a new Primary / Secondary widget to the dashboard. The widget lists the primary tickets and the secondary tickets of each primary ticket. The agent can hover the mouse over the number of secondary tickets to display a popup window that lists the secondary tickets.

Using the new widget is the same as the usage of other widgets on the dashboard.

44.3 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

• SILVER, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager

• OTRSProcessManagementArticleEmail

45.1 Administrator Interface

This package has no administrator interface, but the feature add-on adds a new communication channel that is selectable for articles.
45.1.1 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:

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.

6. Save and deploy the process.
7. Go to the Ticket Notifications screen of the administrator interface.
8. Create a corresponding ticket notification for the event ArticleCreate so that the email is sent.
Warning: In the same activity dialog, you can use either the above-mentioned article field with email or a regular article field. Using both fields together will cause a process error.

Now go to the Tickets and start a new process.

45.2 Agent Interface

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

45.2.1 Tickets

After installation of the package, it is possible to send an article to a customer user when an agent starts a new process ticket.

New Process Ticket

To send an email to a customer user via a process ticket:

1. Click on the Process Ticket menu item in the Action menu or click on the Start Processes menu item in the main menu.
2. Select a process from the drop-down list.
3. Enter your customer information as already known from email tickets.

On submitting the activity dialog, the article gets sent to the configured customer user.

45.3 External Interface

This package has no external interface.
Warning: This feature add-on has reached the end of life with OTRS 7 and will not be longer supported.

The functionality can be easily realized by configuration of generic agent jobs. For existing mappings between queues and users, the generic agent jobs will be automatically created during system upgrade.

New mappings take place in the form of new generic agent jobs by the administrator.

Note: If this feature add-on is already installed, the feature will be replaced by the core functionality of the framework during the upgrading process. After the upgrading process, the package will be removed.
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

• SILVER, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager

• OTRSReady2AdoptITSMProcesses

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

47.1 Administrator Interface

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

47.1.1 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:

<table>
<thead>
<tr>
<th>Service Design::Availability Management (proactive)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Design::Availability Management (reactive)</td>
</tr>
<tr>
<td>Service Design::Catalogue Management</td>
</tr>
<tr>
<td>Service Design::Continuity Management</td>
</tr>
<tr>
<td>Service Design::Service Design Coordination (individual design)</td>
</tr>
<tr>
<td>Service Design::Service Level Management</td>
</tr>
<tr>
<td>Service Design::Supplier Management</td>
</tr>
<tr>
<td>Service Operation::Access Management</td>
</tr>
<tr>
<td>Service Operation::Event Management</td>
</tr>
<tr>
<td>Service Operation::Incident Management</td>
</tr>
<tr>
<td>Service Operation::Problem Management</td>
</tr>
<tr>
<td>Service Operation::Request Fulfillment Management</td>
</tr>
<tr>
<td>Service Operation::Service Validation and Testing</td>
</tr>
<tr>
<td>Service Transition::Change Evaluation</td>
</tr>
<tr>
<td>Service Transition::Knowledge Management</td>
</tr>
<tr>
<td>Service Transition::Release and Deployment Management</td>
</tr>
<tr>
<td>Service Transition::Service Asset and Configuration Management</td>
</tr>
<tr>
<td>Service Transition::Service Transition Planning and Support</td>
</tr>
</tbody>
</table>

To import a Ready2Adopt process:
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.

### 47.2 Agent Interface

This package has no agent interface.

### 47.3 External Interface

This package has no external interface.
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, GOLD, TITANIUM, PLATINUM

**Package Name in OTRS Package Manager**

- OTRSReady2AdoptProcesses
48.1 Administrator Interface

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

48.1.1 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:

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.

48.2 Agent Interface

This package has no agent interface.

48.3 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, TITANIUM, PLATINUM

**Package Name in OTRS Package Manager**

- OTRSReady2AdoptWebServices

---

**Note:** This feature add-on requires the *Ticket Invoker* feature.
49.1 Administrator Interface

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

49.1.1 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.

49.2 Agent Interface

This package has no agent interface.

49.3 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, GOLD, TITANIUM, PLATINUM

**Package Name in OTRS Package Manager**

- OTRSRestorePendingInformation

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

- Ticket Forms
50.1 Administrator Interface

This package has no administrator interface.

50.2 Agent Interface

After installation of the package, the functionality is automatically integrated in the Set Pending Time action. If there was an unexpired pending time set before, the functionality pre-fills the fields with the restored information from the last pending reminder set by the agent. This includes the following fields, which are pre-filled in the Set Pending Time action:

- State
- Pending Time
- Subject
- Body (if the text comes from a note article created by the Set Pending Time action)

Note: The functionality works only, when the pending time is not expired. When the pending time is expired the standard functionality sets the usual default values.

50.2.1 Use Case Example

1. The agent sets a pending state with a pending time and other information like subject and body text in the ticket.
2. The pending time is stored now in the property card of the Properties widget in the ticket detail view.
3. Before the pending time is expired, an incoming customer communication resets the pending time (the customer answers with “thanks”).
4. With the functionality of this package the agent is able to restore the pending time within the Set Pending Time action easily.
5. When the agent opens the Set Pending Time action, the fields are pre-filled with the information from the last entry which set a pending time.

Note: There are several options to set a pending time, but the pre-fill functionality is only integrated in the Set Pending Time action.

Note: The text field is only pre-filled from note articles, since this is the type of article the Set Pending Time action creates.
50.3 External Interface

This package has no external interface.
CHAPTER FIFTYONE

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 kind of defined access possible by assigning customer IDs to partner IDs. It enhances and improves the customer detail view of the standard OTRS version and specifies whose customer data can be accessed by all agents. When customer data is retrieved from any LDAP back end or the local database, the feature add-on checks for a partner ID in the agent’s user data which must origin from an LDAP agent back end; then, only returns those customer users whose customer IDs are assigned to the agent’s partner ID. Therefore, agents of a partner ID can only see customer data of belonging to 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

- SILVER, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager

- OTRSRestrictCustomerDataView
51.1 Administrator Interface

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

51.1.1 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 Screen](image)

**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 the *System Configuration* screen.
2. Select *OTRSRestrictCustomerData View* 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 the *System Configuration* screen.
2. Select *OTRSRestrictCustomerData View* 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.

![Fig. 2: Add Mapping Screen](image)

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.

![Fig. 3: Edit Mapping Screen](image)

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 Table]

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.

---

**51.2 Agent Interface**

This package has no agent interface.

---

**51.3 External Interface**

This package has no external interface.
SAP SOLUTION MANAGER CONNECTOR

This package includes extension modules for the generic interface that provides two-way communication and ticket synchronization between OTRS and SAP Solution Manager.

Note: SAP Solution Manager (reachable via local or remote network) is required for this feature.

Available in Service Package
- SILVER, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager
- OTRSSAPSolutionManagerConnector

52.1 Administrator Interface

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

52.1.1 Bidirectional Communication

The modules included in this package provide communication between OTRS and a SAP Solution Manager system (from now on referred to as SolMan in this document), however it is necessary to first setup both systems independently in order make sure that both systems work correctly and then execute all the necessary steps in the SolMan system to request and provide actions trough a non-SAP web service.

With this package (and the necessary customized configuration) OTRS will be able to communicate with an existing SolMan by pretending to be another SolMan system, using the same function names, data structures and value types.

OTRS does not implement the full set of SolMan API. Please refer to the Web Services chapter to check which ones are available in OTRS.

Prior to any attempt to creating an OTRS web service with this package modules is necessary to know how SolMan handle concepts like priorities, states (SAPUserStatus), article types (TextTypes), etc. and the values for this concepts in the used instance of SolMan system, as well as in OTRS.

OTRS and the SolMan system should be able to communicate over the network to send and receive SOAP messages between them. Please be sure to open the necessary ports on both servers and any network device between them (like a switch or firewall).

In order to use this package and the contained modules is necessary to create or modify a web service within OTRS. Please refer to the Web Services chapter for more information.
52.1.2 Automatic Creation of Customer Users and Agents in OTRS

Tickets exchanged between OTRS and SolMan also transfer ticket owner and customer user.

If an incoming ticket creation/update contains data of an unknown agent, a new agent will be automatically created in OTRS. The same behavior exists similarly for customer users.

52.1.3 Multiple SolMan Instances

It is possible to connect to multiple SolMan instances from within the same OTRS system. Just use one web service per instance.

The system GUID of each system is stored separately and so is the ticket synchronization state.

Processes & Automation

This package makes it possible to connect OTRS with SolMan using web services.

Web Services

As mentioned in the feature list, a basic knowledge of both OTRS and SolMan web services is assumed.

An example web service is provided with the package in doc/SolManConnectorExample.yml. It might be a good idea to keep this web service installed (but disabled) as a duplicate on your system as a reference.

For necessary and recommended modifications of each web service part see sections below. Modifications refer to the provided example.

Requester Transport

Be sure to use HTTP::SOAPSolMan transport. This is necessary for correct namespace generation.

Set the endpoint (URI) of your SolMan instance and set authentication, proxy and SSL options if necessary.

The SOAPAction options, namespace and name scheme should not be changed.

Requester Invokers

This package includes three different invoker controllers that provide functionality for all supported function calls.

- The SolMan::IncidentCreate controller handles initial transfer of a ticket to SolMan (e.g. ProcessIncident, ReplicateIncident).
- The SolMan::IncidentUpdate controller handles follow-up transfer of a ticket to SolMan (e.g. AcceptIncidentProcessing, AddInfo, CloseIncident, RejectIncidentSolution, VerifyIncidentSolution).
- The SolMan::RequestSystemGuid controller is used to request system GUID information from SolMan.

It is important to differentiate between IncidentCreate and IncidentUpdate controllers. Otherwise the invoker names do not make a difference handling-wise in OTRS (except CloseIncident which will croak if you try to update an already closed ticket).

If an update-type invoker is triggered for a ticket without any previous sync activity, they will automatically trigger a create type invoker (ProcessIncident if it exists or alternatively ReplicateIncident).
The invoker for requesting the SolMan system GUID is triggered automatically whenever this information is not known in OTRS.

Every regular invoker will transfer all existing articles that have not been transferred to this point.

By default, the AddInfo invoker will be triggered for every article created in OTRS via an event trigger. It is highly recommended to configure event triggers for all relevant invokers (e.g. for state and priority changes) and to restrict them via event conditions (e.g. to tickets in specific queues).

**Requester Mapping**

Mapping between OTRS and SolMan is done in OTRS via XSLT configuration.

The provided configuration contains working XSLT templates. For standard requirements it is only necessary to modify the outbound mapping template in regard to:

- OTRS file path for XSL helper templates
- OTRS → SolMan state mapping
- OTRS → SolMan priority mapping
- Dynamic field selection and name mapping
- OTRS → SolMan article type mapping

The templates have to be modified individually per invoker but are (by default) identical for all regular invokers.

The data structure provided by the invaders is similar to the structure of the OTRS ticket connector but with some modifications and extra data. This is a quick reference:

```xml
<Customer>
  <!-- customer attributes, e.g. 'UserEmail' -->
</Customer>

<Owner>
  <!-- owner (agent) attributes, e.g. 'UserEmail' -->
</Owner>

<SolMan>
  <IncidentId> incident id </IncidentId>
  <IncidentGuid> incident guid </IncidentGuid>
  <LocalSystemGuid> OTRS system guid </LocalSystemGuid>
  <RemoteSystemGuid> SolMan system guid </RemoteSystemGuid>
</SolMan>

<Ticket>
  <Article> # can occur 0-n times
    <Attachment> # can occur 0-n times
      <!-- attachment attributes, e.g. 'Filename' --> # 'Content' is base64-encoded automatically
    </Attachment>
    <DynamicField> # can occur 0-n times
      <Name> dynamic field name </Name>
      <Value> dynamic field value(s) </Value>
    </DynamicField>
  </Article>
  <!-- article attributes, e.g. 'From' -->
</Ticket>
```

(continues on next page)
The default XSLT template types are provided as individual files for reference (see doc/ExampleMapping-Invoker-*.xsl provided by the package).

Requester Error Handling

Some error handling modules are pre-configured used to reschedule requests if a specific problem occurs. Rescheduled requests are executed in 1 minute intervals for up to 5 tries.

By defaults all transport errors cause a reschedule as well as data preparation where the SolMan system GUID could not be retrieved.

Also requests are rescheduled if SolMan returns an error code, except if this code is 01, 02, 03 or 04.

It is recommended to change retry options to your needs and to check if the error code specific handling matches your requirements.

Provider Transport

To avoid confusion you should use the HTTP::SOAPSolMan transport. Contrary to the requester this is not strictly necessary.

Modify message length if necessary.

The SOAPAction options, namespace and name scheme should not be changed.

Provider Operations

This package includes three different operation controllers that provide functionality for all supported function calls.

- The SolMan::IncidentCreate controller handles initial transfer of a ticket from SolMan (e.g. ProcessIncident, ReplicateIncident).
- The SolMan::IncidentUpdate controller handles follow-up transfer of a ticket from SolMan (e.g. AcceptIncidentProcessing, AddInfo, CloseIncident, RejectIncidentSolution, VerifyIncidentSolution).
- The SolMan::RequestSystemGuid controller is used to provide system GUID information to SolMan.

It is important to differentiate between IncidentCreate and IncidentUpdate controllers. Otherwise the invoker names do not make a difference handling-wise in OTRS.

If a ticket has pending updates that need to be transferred from OTRS to SolMan (e.g. new articles), update from SolMan are temporarily blocked.

Due to the nature of SolMan, the sent attachments cannot be assigned to more than one article. Articles sent within a request will automatically attached to the last article created within this request. If the request does not contain any article, a dummy article will be created for the attachments.
Provider Mapping

Mapping between OTRS and SolMan is done in OTRS via XSLT configuration.

The provided configuration contains working XSLT templates. For standard requirements it is only necessary to modify the inbound mapping template in regard to:

- OTRS file path for XSL helper templates
- OTRS target queue
- SolMan → OTRS state mapping
- SolMan → OTRS priority mapping
- Dynamic field selection and name mapping
- SolMan → OTRS article type mapping

The templates have to be modified individually per operation but are (by default) identical for all regular operations.

The data structure expected by the operations is similar to the structure of the OTRS ticket connector but with some modifications and extra data. This is a quick reference:

```
<Attachment> # can occur 0-n times
    <*> attachment attributes, e.g. 'Filename' </*> # 'Content' is base64-decoded automatically
</Attachment>
<Ticket>
    <*> article attributes, e.g. 'From' </*>  
</Article>
<DynamicField> # can occur 0-n times
    <Name> ticket dynamic field name </Name>
    <Value> ticket dynamic field value(s) </Value>
</DynamicField>
    <*> ticket attributes, e.g. 'TicketNumber' </*> 
</Ticket>
<SolMan>
    <IncidentGuid> incident guid </IncidentGuid>
    <LocalSystemGuid> OTRS system guid </LocalSystemGuid>
    <RemoteSystemGuid> SolMan system guid </RemoteSystemGuid>
    <Person> # can occur 0-n times
        <Meta>
            <ID> OTRS id of agent / or login of customer user </ID>
            <ExternalID> SolMan id of person </ExternalID>
            <Type> CustomerUser, Owner or Agent </Type>
        </Meta>
        <*> agent or customer user attributes, e.g. 'UserEmail' </*> 
    </Person>
</SolMan>
```

The default XSLT template types are provided as individual files for reference (see doc/ExampleMapping-Operation-*.xsl provided by the package).
52.2 Agent Interface

This package has no agent interface.

52.3 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 is configured to make all queues reappear with just one click in the agent interface. The feature is disabled in the external interface by default but can be enabled in the system configuration. 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, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager

- OTRSServiceBasedQueueRouting
53.1 Administrator Interface

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

53.1.1 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.

![Manage Service-Queue Relations Screen](image)

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.

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.
Fig. 2: Change Queue Relations for Service

Fig. 3: Change Service Relations for Queue
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.

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

### Change Field Order on Screens

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.

However, the *Queue* field is before the *Service* field in the screens and this can cause confusion for the users. They select a queue first then they select a service but the selected queue is unselected if the assignment between the queue and the service is not set. The queue selection now depends on the service selection so the service should be the first input field.

To avoid such confusions it is **recommended** to change the field order of the affected screens in the system configuration.

### 53.2 Agent Interface

This package has no agent interface, but it extends the standard behavior of the queue selection method. The service based queue routing affects the ticket creation screens and the *Move Ticket* action. For this, the given service of the selected ticket will be used.

After selecting a service, there is a lookup for configured queues for this service. If there are configured queues for the selected service, only those will be available in the queue selection. If there are no queues configured for the selected service, all queues will be displayed.

It is possible to add a filter button next to the queue selection. After clicking this button all available queues get restored to the queue selection field and can be selected. This feature can be enabled in the system configuration.

Configured ACLs affect the displayed queues as usual.

### 53.3 External Interface

This package has no external interface, but it extends the standard behavior of the queue selection method. The service based queue routing affects the ticket creation screens. For this, the given service of the selected ticket will be used.

After selecting a service, there is a lookup for configured queues for this service. If there are configured queues for the selected service, only those will be available in the queue selection. If there are no queues configured for the selected service, all queues will be displayed.

It is possible to add a filter button next to the queue selection. After clicking this button all available queues get restored to the queue selection field and can be selected. This feature can be enabled in the system configuration.

Configured ACLs affect the displayed queues as usual.
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, GOLD, TITANIUM, PLATINUM

**Package Name in OTRS Package Manager**

- OTRSServiceCategories

### 54.1 Administrator Interface

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

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.

![Manage Type-Service Relations Screen](image)

**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.

**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.
Fig. 2: Change Service Relations for Type

Fig. 3: Change Type Relations for Service
54.2 Agent Interface

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

54.2.1 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*.

54.3 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

• SILVER, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager

• OTRSSpecificTicketNotifications

55.1 Administrator Interface

This package has no administrator interface, but extends Ticket Notifications screen with new buttons and adds a new group to the system.
55.1.1 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.

![Group Management Screen](image)

**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 Create Notification button in the Ticket Notifications widget of the Personalization menu.

**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*
55.1.2 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.

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.

![Recipients](image)

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.

55.2 Agent Interface

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.

55.2.1 Personalization

This package extends the Ticket Notifications widget in the Notification Settings menu item of the Personalization menu.

Notification Settings

Every agent who is a member of the group custom_notifications is able to create specific ticket notifications. The Create Notification button in the bottom of the Ticket Notifications widget points to the Ticket Notifications screen of the administrator interface.

To add a personal ticket notification:

1. Click on the Create Notification button and login to the administrator interface with your password.
2. Click on the Add Notification button as an agent or on the Add personal notification button as an administrator in the left sidebar if the Add Notification screen is not opened.
3. Fill in the required fields.
4. Click on the Save button.

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

To edit a personal ticket notification:
Ticket Notifications

- Supervisor
- Ticket create notification
- Ticket follow-up notification (locked)
- Ticket follow-up notification (unlocked)
- Ticket lock timeout notification
- Ticket queue update notification
- Ticket service update notification

Create Notification

Fig. 4: Ticket Notifications Widget

Ticket Notification Management

<table>
<thead>
<tr>
<th>Actions</th>
<th>List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Notification</td>
<td>Supervisor</td>
</tr>
</tbody>
</table>

Fig. 5: Ticket Notification Management Screen
1. Click on a ticket notification in the list of ticket notifications. If you are an administrator, the personal ticket notifications are displayed in orange rows.

2. Modify the fields.

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

Alternatively the agent can use the gear icon in the ticket notification list from the agent interface. This icon provides a direct link to the specific ticket notification to be edited.

To delete a personal ticket notification:

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

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.

### 55.3 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, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager

- OTRSStatePreselectionResponseTemplates

56.1 Administrator Interface

After installation of the package a new module will be available in the Ticket Settings group of the administrator interface.
56.1.1 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](image)

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](image)

Fig. 2: Edit Response Screen

**Note:** The configured ticket states must be allowed in ticket context as next state. If this should not be the case, no next state will be pre-selected when answering a ticket.

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

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

**Pre-selected ticket state**

Select a ticket state to assign it to the template. This ticket state will be pre-selected on reply actions when answering a ticket using this response template.

56.2 Agent Interface

This package has no agent interface, but the package allows pre-selection of ticket states for response templates.

To pre-select a ticket state on reply:

1. Go to the ticket detail view and click on the *Reply via Email* action in the article menu.
2. Select a response template from the drop-down list.
3. The pre-selected ticket state should be set in the *State* field of the reply action.

**See also:**

The ticket state can be assigned in the *State Pre-selection for Templates* management screen.

56.3 External Interface

This package has no external interface.
This module enables you to create surveys, add questions to a created survey, set a survey live in order to send requests, modify existing surveys to a certain extent and view statistics of submitted survey answers as well as the answers themselves.

The surveys can be sent to customer users of closed tickets. The results are displayed as graphical statistics of completed surveys.

**Available in Service Package**

- SILVER, GOLD, TITANIUM, PLATINUM

**Package Name in OTRS Package Manager**

- Survey

### 57.1 Administrator Interface

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

#### 57.1.1 Administration

After installation of the package a new module will be available in the administrator interface.

**Surveys**

Use this screen to add surveys to the system. The survey management screen is available in the *Surveys* module of the *Administration* group.

![Survey Overview Screen](image)

Fig. 1: Survey Overview Screen
Manage Surveys

To add a survey:
1. Click on the Add new survey button in the header bar.
2. Fill in the required fields.
3. Click on the Create button.
4. Click on the Edit Questions button in the header bar.
5. Add some questions and click on the Add button.
6. Click on the Close this window button to go back.

**Warning:** Surveys cannot be deleted from the system. They can only be deactivated by setting the Validity option to invalid.

To edit a survey:
1. Click on a survey in the list of surveys.
2. Click on the Edit General Information button in the header bar.
3. Modify the fields.
4. Click on the Update button.

To filter the surveys displayed in the overview table:
1. Click on the Survey overview options button in the header bar.
2. Define some filters or date restrictions.
3. Click on the Submit button.

To make the survey available to use:
1. Click on a survey in the list of surveys.
2. Change the status of the survey.

**Warning:** As soon as the survey has the state Master or Valid the questions cannot be edited anymore.

Survey Settings

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

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

**Introduction** *
- This is the summary of the survey. It is displayed in the external interface together with the title.

**Sender** *
- This is the email address which is set as sender of the email sent to the users.
Subject *
The subject of the email sent to the users.

Body *
The body text of the message.

Queues
Select the queues for which the new survey can be created. If none is selected, it will be valid for every queue.

Ticket Types
Select the ticket types for which the new survey can be created. If none is selected, it will be valid for every ticket type.

See also:
This field is not displayed by default. It can be enabled in the system configuration.

Services
Select the services for which the new survey can be created. If none is selected, it will be valid for every service.

See also:
This field is not displayed by default. It can be enabled in the system configuration.

Internal 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.

Customer conditions
It is possible to define conditions based on customer user attributes, e.g. all customer users with a specified comment would not get any survey emails. Each attribute could have at least one condition. A condition contains a checkbox value for negation and a value field (input type text or multiple select).

See also:
This field is not displayed by default. It can be enabled in the system configuration.

Add Questions

As soon as a new survey is created, the user can add questions to the survey by clicking on the Edit Questions menu item in the displayed survey. A new window will pop up where the user can add questions.

Five types of questions are selectable. These are:

- **Yes/No questions**: a user can answer by selecting Yes or No from a drop-down field.
- **Radio (List) questions**: a user can answer by selecting exactly one and just one of the possible answers.
- **Checkbox (List) questions**: a user can answer by selecting none to all listed answers.
- **Textarea questions**: a user can write answers by typing regular text.
- **Net Promoter Score questions**: a user can answer by selecting exactly one and just one of the possible scores.
- **Embedded question**: a user can answer directly in the email received. The possible answers can be images embedded in the email that is sent out. Only one embedded question can be added to a survey. The embedded question will not be shown in the external interface.

For each answer type you can choose if you want a customer user to be required to answer this question or if a customer user can ignore the question.
As soon as questions of type Radio (List), Checkbox (List) or Net Promoter Score got added, the question will be shown in the Survey Questions overview. By clicking on the question you can add answers, change the answer order by clicking on the up or down icon or delete the answer by clicking on the trash icon.

Questions of type Radio (List) and Net Promoter Score needs to have at least two answers to be valid. Furthermore, questions of type Checkbox (List) needs to have at least one answer to be valid.

After all questions and possible answers have been added, you can leave the Edit Questions area by clicking on the Close this window button.

Start Sending Survey Requests

On the survey detail page, you can select Master from the - Change Status - drop-down. As soon as the survey has the state Master the survey is set live.

Note: Only one survey can have the state Master.

If a ticket is closed, the customer user receives an email invitation to fill the survey by clicking a link inside the invitation email.

Survey Results Graph

Select a survey that is already submitted to the customer users and scroll down. You can see the Survey Results Graph widget. This displays statistically a summary of questions like Yes/No, Radio (List) and Checkbox (List).

Questions where answers were configured as required will have one vote for each submitted survey by a customer user.

Questions where answers were not configured as required are optional for the customer users. If customer users have not selected any of the Radio (List), Yes/No or Checkbox (List) answers or have not entered anything in a Textarea answer, it can happen that only 10 or 20 answers appear in a statistic, although the survey was sent by 200 customer users.

Statistics for required Checkbox (List) questions have at least one checked value for each answer submitted. However, since multiple selections are possible, there may be more answers than the total number of surveys submitted.

Example: 200 surveys submitted by customer users, 350 answers on one Checkbox (List) question.

Statistics for not required questions may have less answers than the total amount of submitted surveys (answer is optional).

Statistics Details

When you click on a survey that customer users have already submitted, you can see the following details in the Stats Details section:

- The time when the survey invitation was sent to the customer user.
- The time when the customer user submitted the filled survey.
- The associated ticket as a link.
- A magnifying glass showing the votes selected for that customer user.

In this area, you will find the answers to each question that the customer user has selected or typed in. Especially Textarea answers are of interest here.
Modifying Survey

As soon as a survey had the status *Master*, the modification possibilities are limited. Therefore, make sure that the number of questions and answers is configured correctly before changing the status to *Master*.

**Warning:** In a survey of type *Master* you can not add or delete questions any more.

In a survey of type *Master* the general information (title, introduction, sender etc.), the question text, the answer text, as well as the order of the answers can be changed.

You can change the *Answer Required* property of a question too, but this may just make sense as long as there are no submitted customer surveys. Changing the *Answer Required* question to *No* makes the answer optional, so the statistic for this question may not have an answer for each submitted customer survey any more. Changing the *Answer Required* property to *Yes* will require the customer users of the future survey to give a vote for this question, but will not add votes for already submitted surveys. So statistics will render rather useless too.

### 57.2 Agent Interface

This package has no agent interface.

### 57.3 External Interface

This package displays a customer survey on the external interface. The survey is sent to the customer users of closed tickets via email, but the survey is also accessible from the 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, GOLD, TITANIUM, PLATINUM

**Package Name in OTRS Package Manager**

- OTRSSystemConfigurationHistory

### 58.1 Administrator Interface

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

### 58.1.1 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.

![Deployment History Screen](image)

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 Screen](image)

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.

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.
Fig. 4: Change History Screen
58.2 Agent Interface

This package has no agent interface.

58.3 External Interface

This package has no external interface.
This module implements a basic interface to system monitoring suites.

Note: A network monitoring system, such as Nagios, Icinga2, HP OpenView or similar, capable of sending out events via email is required to use this package.

For Nagios, it works by receiving email messages sent by a network monitoring suite. New tickets are created in case of component failures. Once a ticket has been opened messages regarding the effected component are attached to this ticket. When the component recovers, the ticket state can be changed or the ticket can be closed. If an open ticket for a given host and service combination exists, all mails concerning this particular combination will be attached to the ticket until it’s closed.

For Icinga2, it works by specifying a host and a service in ticket dynamic fields. This combination of host and service is used after a ticket lock is set to generate a HTTP request which is send to the configured Icinga2 host. In Icinga2, this request is used to create or to confirm new incidents.

Available in Service Package

- SILVER, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager

- SystemMonitoring

59.1 Administrator Interface

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

59.1.1 Control Flow for Nagios

The following control flow illustrates how mails are handled by this module and in which cases they trigger which action. Pretty much all checks are configurable using the regular expressions given by the parameters listed below.

- Mail matches FromAddress?
  - NO → Continue with regular mail processing
  - YES → Does a ticket with matching host and service combination already exist in OTRS?
    - NO → Does State: match NewTicketRegExp?
      - NO → Stop processing this mail (silent drop)
      - YES → Create new ticket, record host and service, attach mail
    - YES → Continue with regular mail processing

∗ YES → Attach mail to ticket. Does State: match CloseTicketRegExp?
  ∗ NO → Continue with regular mail processing
  ∗ YES → Change ticket type as configured in CloseActionState

Besides of a few additional sanity checks this is how the system monitoring module treats incoming mails. By changing
the regular expressions it should be possible to adopt it to different monitoring systems.

59.1.2 Icinga2 Acknowledge

A new ticket is created with values in the specified dynamic fields which are needed as combination of host and service
for the communication to the Icinga2 host. After this newly created ticket is locked to an agent, a HTTP request is send
to the configured Icinga2 host. In the Icinga2 host a new acknowledge is created or an existing one is confirmed.

Administration

Any system requires configuration. Configuring a system should be an easy task and the tools for configuration fit-for-
purpose.

System Configuration

This package contains two useful functions, which can be used to automatize the workflow based on the messages from
the system monitoring suites. However, both functions are turned off by default, and they can be enabled in the system
configuration.

Note: The OTRS Configuration Management feature is required to use these functions.

Link Incident Ticket with Configuration Item

An already opened incident ticket can be linked with the affected configuration item. This is only possible when a subse-
quent system monitoring email arrives.

The setting SystemMonitoring::LinkTicketWithCI needs to be enabled to use this feature.

Set Incident State

It is possible to set the incident state of a configuration item automatically when a system monitoring email arrives.

The setting SystemMonitoring::SetIncidentState needs to be enabled to use this feature.
Allow Program Safe to Run

External programs to be run by OTRS are blocked by default due to security reasons. You have to add the program to the allow list as described in the Administration Manual.

Processes & Automation

After installation of the package some new dynamic fields will be added to the system, which are required for working this package properly.

Dynamic Fields

This package adds three new dynamic fields to the system.

**Warning:** Do not delete these dynamic fields. They are required to work the system monitoring package properly.

New Dynamic Fields

**SystemMonitoring HostName**

This dynamic field will store the host name comes from the system monitoring suit.

- **Internal name:** TicketFreeText1
- **Type:** Text
- **Object:** Ticket

**SystemMonitoring ServiceName**

This dynamic field will store the service name comes from the system monitoring suit.

- **Internal name:** TicketFreeText2
- **Type:** Text
- **Object:** Ticket

**SystemMonitoring StateName**

This dynamic field will store the state name comes from the system monitoring suit.

- **Internal name:** ArticleFreeText1
- **Type:** Text
- **Object:** Article
59.2 Agent Interface

This package has no agent interface, but in case the incident ticket linking with configuration item and the incident state setting are enabled in the System Configuration, the list of configuration items and the list of services can be changed based on the incident tickets.

For more information, read the Services and the Configuration Items chapters in the OTRS::ITSM manual.

59.3 External Interface

This package has no external interface.
Warning: This feature add-on has reached the end of life with OTRS 7 and will not be longer supported.
CHAPTER SIXTYONE

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

- SILVER, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager

- OTRSTicketAllocation

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

- Advanced Generic Agent
61.1 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 Agent 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.

61.1.1 Ticket Settings

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

Competence Levels

Use this screen to add competence levels to automatic allocation. The competence level management screen is available in the Competence Levels module of the Ticket Settings group.

Fig. 1: Competence Level Management Screen
Manage Competence Levels

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

To add a competence level:
1. Click on the Add Competence Level button in the left sidebar.
2. Fill in the required fields.
3. Click on the Save button.

![Add Competence Level Screen](Fig. 2: Add Competence Level Screen)

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

Note: It’s recommended to limit your system to 5 competence levels or less and reuse the current 5 to keep the use of the ticket allocation system.

To edit a competence level:
1. Click on a competence level in the list of competence levels.
2. Modify the fields.
3. Click on the Save or Save and finish button.

![Edit Competence Level Screen](Fig. 3: Edit Competence Level Screen)
Note: If several competence levels are added to the system, use the filter box to find a particular competence level by just typing the name to filter.

Competence Level 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.

**Level** *

The level of the competence.

**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.

61.1.2 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.

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 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
**Events**

**Event Triggers:**

<table>
<thead>
<tr>
<th>TYPE</th>
<th>EVENT</th>
<th>ALL-TICKETS</th>
<th>DELETE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ticket</td>
<td>TicketCreate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ticket</td>
<td>TicketQueueUpdate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Session</td>
<td>SessionCreate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Session</td>
<td>SessionRemove</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ticket</td>
<td>TicketStateUpdate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ticket</td>
<td>TicketLockUpdate</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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:**

- Ticket
- EscalationResponseTimeNotifyBefore

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

**Event execution:**

- Daemon

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:**

- Yes

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

Fig. 4: Events Section
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.

### 61.1.3 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.

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.
### Competences Screen

**Group competence(s)**
- **Group** competence(s):
  - admin: very low
  - users: very high

**Priority competence(s)**
- **Priority** competence(s):

**Queue competence(s)**
- **Queue** competence(s):
  - Misc: very high

**Type competence(s)**
- **Type** competence(s):

---

Fig. 6: Competences Screen
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.

## 61.2 Agent Interface

Agents are able to edit their own competences, if this is enabled in the system configuration.

### 61.2.1 Personalization

Agents are able to edit their own competences, if this is enabled in the system configuration.

**Competences**

This menu item contains the same settings for the agent, that a system administrator can set for the Agents.

---

**Note:** The system configuration `OTRSTicketAllocation::AllowAgentEditOwnCompetences` has to be enabled to access this menu item from the Personalization.

## 61.3 External Interface

This package has no external interface.
**Competences**

Start typing to filter...

**Group competence(s)**

admin

1 very low

users

5 very high

Add New

Please select

**Priority competence(s)**

Add New

Please select

**Queue competence(s)**

Misc

5 very high

Add New

Please select

**Type competence(s)**

Add New

Please select

---

Fig. 7: Competences Selection
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 \(<\text{OTRS_CONFIG\_*}>\) and \(<\text{OTRS_CURRENT\_*}>\) in the template body.

With this feature add-on the following screens can be configured:

- **Agent interface**
  - New phone ticket
  - New email ticket
- **External interface**
  - New ticket

In addition, the included feature add-on *Hide/Show Dynamic Fields*, which is also available as a separate feature add-on, enables you to define which dynamic fields you want to display in your ticket forms and which ones you want to hide. This gives you more freedom to configure all ticket screens to your specific needs.

Another included feature add-on is *Restore Pending Information*, which makes it possible to restore the pending reminder information of the last reminder when you open a new reminder in the ticket pending pop-up screen.

The following information can be restored:

- Pending date
- State
- Title
- Text (if the text comes from a note article created by the *Set Pending Time* action)

This feature add-on works when the target time of the reminder has not expired. Otherwise the standard functionality sets the usual default values.

**Benefits**

- Flexible reaction to a variety of customer requests.
- Acceleration of processes by using forms.
- Increases the completeness of forms by defining mandatory fields.

**Target Groups**
62.1 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.

62.1.1 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.
**Manage Dynamic Ticket Templates**

**Note:** To add attachments to a dynamic ticket template, it needs to create the attachment first in the *Attachment Management* screen.

To add a dynamic ticket template:

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

To edit a dynamic ticket template:

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

To delete a dynamic ticket template:

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

**Note:** If several dynamic ticket templates are added to the system, use the filter box to find a particular dynamic ticket template by just typing to filter.
Fig. 2: Add Dynamic Ticket Template Screen
### Edit Template

**General:**
- **Name:** VWTemplate1
- **Comment:** VW Template 1
- **Valid:** valid
- **Frontend:** Agent and External Interface

**Restrictions:**
- **Type:** Unclassified
- **Service:** VWService

**Template:**
- **Workflow:** VWWorkflow1
- **Subject:** VW Service Request 1

**Body:**
- Oil Change
- Oil Filter Change
- Air Filter Change
- Fluids Check

**Attachments:**
- **Brand:** Hide
- **VW Model:** Show as mandatory
- **VW Production Facility:** Show
- **Peugeot Model:** Hide
- **Peugeot Production Facility:** Hide
- **Fuel:** Hide
- **Accessories:** Hide
- **Remarks:** Hide
- **Registration Date:** Hide
- **Invoice Date:** Hide

[Save] or [Save and finish] or [Cancel]
Dynamic Ticket Template Settings

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

Name *

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

Comment

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

Valid *

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

Frontend *

Defines, where can the dynamic ticket template be used. A dynamic ticket template can be used in the following interfaces:

- Agent Interface
- Agent and External Interface
- External Interface

Type *

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

Service *

Select a service, that has been added to the system.

Subject

The subject of the email sent to the users.

Body

The body of the email sent to the users.

Attachments

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

Fig. 4: Delete Dynamic Ticket Template Screen
Dynamic Fields

The list of dynamic fields are displayed in this section. Choose for each dynamic field to *Hide*, *Show* or *Show as mandatory* on ticket create screens.

Dynamic Ticket Template Variables

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

For example, the variable `<OTRS_CURRENT_UserFullname>` expands to the full name of the current user allowing a template to include something like the following:

```
From: <OTRS_CURRENT_UserFullname>
```

This tag expands, for example to:

```
From: John Smith
```

62.1.2 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 the 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.

62.1.3 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.

Fig. 8: Change Dynamic Ticket Template Relations for Group

**Note:** If several dynamic ticket templates or groups are added to the system, use the filter box to find a particular dynamic ticket template or group by just typing the name to filter.

Multiple dynamic ticket templates or groups can be assigned in both screens at the same time. Additionally clicking on a dynamic ticket template or clicking on a group in the relations screen will open the Edit Template screen or the Edit Group screen accordingly.
**Warning:** Accessing a dynamic ticket template or a group provides no back link to the relations screen.

### 62.1.4 Example Usage

For this example we will use the same fields as in the example of *Hide/Show Dynamic Fields*, but is necessary to comment or remove the ACLs.

**Note:** While the dynamic ticket templates can work in conjunction with other ACL rules, it is recommended to test the templates without other affecting ACLs, and after the templates work as needed, incorporate other ACLs one by one.

#### Preparation

To show the full functionalities of the dynamic ticket templates we can simply remove all dynamic fields from the *New Phone Ticket* screen by resetting the system configuration setting `Forms###AgentFrontend::TicketCreate::Phone::CreateProperties` to the default value, or disable the specific dynamic fields in the YAML configuration of this form:

```yaml
- Label: Dynamic Fields
  Collapsible: 1
  Fields:
    - Name: DynamicField_Brand
      Inactive: 1
    - Name: DynamicField_VWModel
      Inactive: 1
    - Name: DynamicField_VWProductionFacility
      Inactive: 1
    - Name: DynamicField_PeugeotModel
      Inactive: 1
    - Name: DynamicField_PeugeotProductionFacility
      Inactive: 1
    - Name: DynamicField_Fuel
      Inactive: 1
    - Name: DynamicField_Accessories
      Inactive: 1
    - Name: DynamicField_Remarks
      Inactive: 1
    - Name: DynamicField_RegistrationDate
      Inactive: 1
    - Name: DynamicField_InvoiceDate
      Inactive: 1
```

To define a dynamic ticket template it is necessary to specify a ticket type and a service. If these features are not activated by default, please follow the next steps before start using this feature.

To activate ticket type and service features:

1. Go to the *System Configuration* screen.
2. Select *OTRS* in the *Navigation* widget.
3. Navigate to *Core → Ticket* in the navigation tree.
4. Scroll down to the setting `Ticket::Service` and enable it.
5. Scroll down to the setting `Ticket::Type` and enable it.

**Note:** If you have `OTRSServiceManagement` package installed, these options are already active.

To activate ticket type and service features for the external interface:

1. Go to the `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:

<table>
<thead>
<tr>
<th>Subject</th>
<th>VW Service Request 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Body</td>
<td></td>
</tr>
<tr>
<td></td>
<td>– Oil Change</td>
</tr>
<tr>
<td></td>
<td>– Oil Filter Change</td>
</tr>
<tr>
<td></td>
<td>– Air Filter Change</td>
</tr>
<tr>
<td></td>
<td>– Fluids Check</td>
</tr>
</tbody>
</table>

  | Dynamic Fields |                      |
  |               | – VWModel: visible and required |
  |               | – VWProductionFacility: visible |

- If the selected service is **VW Service** and the selected dynamic ticket template is **VW Template 2**, the ticket form should look like:

<table>
<thead>
<tr>
<th>Subject</th>
<th>VW Service Request 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Body</td>
<td>The selected accessory reports the following issues:</td>
</tr>
<tr>
<td></td>
<td>– VWModel: visible</td>
</tr>
<tr>
<td></td>
<td>– Accessories: visible and required</td>
</tr>
<tr>
<td></td>
<td>– Remarks: visible</td>
</tr>
<tr>
<td></td>
<td>– RegistrationDate: visible</td>
</tr>
<tr>
<td></td>
<td>– InvoiceDate: visible</td>
</tr>
</tbody>
</table>

  | Dynamic Fields |                      |
  |               | – 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:

<table>
<thead>
<tr>
<th>Subject</th>
<th>Peugeot Service Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>Body</td>
<td>The car computer reports the following issues:</td>
</tr>
<tr>
<td>Dynamic Fields</td>
<td>– PeugeotModel: visible</td>
</tr>
<tr>
<td></td>
<td>– PeugeotProductionFacility: visible</td>
</tr>
<tr>
<td></td>
<td>– Fuel: visible</td>
</tr>
</tbody>
</table>

### Create Dynamic Ticket Templates

To create a dynamic ticket template:

1. Open the *Dynamic Ticket Templates* module of the *Ticket Settings* group in the administrator interface.
2. Click on the *Add template* button in the left sidebar.
3. Fill in the required fields.
4. Click on the *Save* button.

If you have added other dynamic fields by your own, the template form will show more fields than the ones explained in this examples. There is nothing to worry about, just leave all other dynamic fields as *Hide* while you create the new templates.

There are special dynamic fields from other packages that are not displayed in the template form. These special fields have a determined behavior and they should be always hidden, always shown or their visibility depends on other configurations, so they can not be part of the template definition. Also any dynamic field marked as *Internal* will not be displayed in the template form.

**See also:**

A quick way to know if a dynamic field is *Internal* is to look at dynamic fields in the overview table of the *Dynamic Fields Management* screen. *Internal* dynamic fields can not be deleted, and they do not have a icon in the *Delete* column.

Create the *VW Template 1* dynamic ticket template with the following data:
Create the **VW Template 2** dynamic ticket template with the following data:

<table>
<thead>
<tr>
<th>Field name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>VWTemplate2</td>
</tr>
<tr>
<td>Comment</td>
<td>VW Template 2</td>
</tr>
<tr>
<td>Valid</td>
<td>valid</td>
</tr>
<tr>
<td>Frontend</td>
<td>Agent and External Interface</td>
</tr>
<tr>
<td>Type</td>
<td>Unclassified</td>
</tr>
<tr>
<td>Service</td>
<td>VW Service</td>
</tr>
<tr>
<td>Subject</td>
<td>VW Service Request 2</td>
</tr>
<tr>
<td>Body</td>
<td>• Oil Change</td>
</tr>
<tr>
<td></td>
<td>• Oil Filter Change</td>
</tr>
<tr>
<td></td>
<td>• Air Filter Change</td>
</tr>
<tr>
<td></td>
<td>• Fluids Check</td>
</tr>
<tr>
<td>Attachments</td>
<td></td>
</tr>
<tr>
<td>Brand</td>
<td>Hide</td>
</tr>
<tr>
<td>VWModel</td>
<td>Show as mandatory</td>
</tr>
<tr>
<td>VWProductionFacility</td>
<td>Show</td>
</tr>
<tr>
<td>PeugeotModel</td>
<td>Hide</td>
</tr>
<tr>
<td>PeugeotProductionFacility</td>
<td>Hide</td>
</tr>
<tr>
<td>Fuel</td>
<td>Hide</td>
</tr>
<tr>
<td>Accessories</td>
<td>Show as mandatory</td>
</tr>
<tr>
<td>Remarks</td>
<td>Hide</td>
</tr>
<tr>
<td>RegistrationDate</td>
<td>Hide</td>
</tr>
<tr>
<td>InvoiceDate</td>
<td>Hide</td>
</tr>
</tbody>
</table>

Create the **Peugeot Template** dynamic ticket template with the following data:

<table>
<thead>
<tr>
<th>Field name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>PeugeotTemplate</td>
</tr>
<tr>
<td>Comment</td>
<td>Peugeot Template</td>
</tr>
<tr>
<td>Valid</td>
<td>valid</td>
</tr>
<tr>
<td>Frontend</td>
<td>Agent and External Interface</td>
</tr>
<tr>
<td>Type</td>
<td>Unclassified</td>
</tr>
<tr>
<td>Service</td>
<td>VW Service</td>
</tr>
<tr>
<td>Subject</td>
<td>Peugeot Service Request</td>
</tr>
<tr>
<td>Body</td>
<td>The selected accessory reports the following issues:</td>
</tr>
<tr>
<td>Attachments</td>
<td></td>
</tr>
<tr>
<td>Brand</td>
<td>Hide</td>
</tr>
<tr>
<td>VWModel</td>
<td>Show</td>
</tr>
<tr>
<td>VWProductionFacility</td>
<td>Hide</td>
</tr>
<tr>
<td>PeugeotModel</td>
<td>Hide</td>
</tr>
<tr>
<td>PeugeotProductionFacility</td>
<td>Hide</td>
</tr>
<tr>
<td>Fuel</td>
<td>Hide</td>
</tr>
<tr>
<td>Accessories</td>
<td>Show as mandatory</td>
</tr>
<tr>
<td>Remarks</td>
<td>Show</td>
</tr>
<tr>
<td>RegistrationDate</td>
<td>Show</td>
</tr>
<tr>
<td>InvoiceDate</td>
<td>Show</td>
</tr>
</tbody>
</table>
Activate Dynamic Ticket Templates

Once there are one or more templates activated in the system, you can use them.

**Note:** ACL restrictions will be ignored for the superuser account (UserID 1).

To activate a dynamic ticket template:

1. Click on the *Take Phone Call* menu item of the *Action* menu in the header bar.
   
   There is a new field called *Dynamic Ticket Template*. By default it is empty and it will be filled automatically when you select the appropriate combination of ticket type and service.

2. Fill the form with the following data:
   
   • Type: Unclassified
   • Customer user: Customer 1
   • Queue: Misc
   • Service: VW Service

3. The *Dynamic Ticket Template* field will be automatically populated with *VW Template 1* and *VW Template 2*.

4. Select each template and match the resulting *New Phone Ticket* form with the expected results.

5. Now change the selected service to *Peugeot Service*. As there is only one template defined for this combination of ticket type and service, the template will be automatically selected for you.

6. Compare the resulting *New Phone Ticket* form with the expected results.

This complete example can be tested also *New Email Ticket* or external *New Ticket* screen without changing anything.
The template definition is independent from the screen and once a template is defined, it can be used on any ticket creation screen, but please review the dynamic field configuration on each screen for the default fields configuration when a template is not selected.

**Dynamic Ticket Templates Customer Groups**

For this example we will use the *Peugeot Template* defined in the example above.

The goal to have a pre-filled form to create a new ticket in external interface.

To configure a customer group:

1. Go to the *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 the *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.

### 62.2 Agent Interface

This package has no agent interface.

**See also:**

There is a detailed use-case example in *Administrator Interface* chapter.
62.3 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.

In addition, the package follows HTTP redirect responses (HTTP code 301, 307 and 308) for outbound communication of HTTP::REST and HTTP::SOAP based web services.

Further the package allows to send empty body HTTP requests (POST, PUT, PATCH) for outbound communication of HTTP::REST based web services.

**Available in Service Package**

- SILVER, GOLD, TITANIUM, PLATINUM

**Package Name in OTRS Package Manager**

- OTRSTicketInvoker

**63.1 Administrator Interface**

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

**63.1.1 Processes & Automation**

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

**Web Services**

The actions that can be performed when you are using OTRS as a requester are called **invokers**. Each invoker belongs to a controller (controllers are collections of operations or invokers). Usually invokers from the same controller need similar settings and share the same configuration dialogs. Each invoker can have independent configuration dialogs if needed.

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:

```php
my $Result = $InvokerObject->PrepareRequest(
    Data => {
        TicketID => 123, # data payload
        ArticleID => 123,
    },
);
```

Invoker result:

```json
{
    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 => {
                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', # optional
            # ContentType or MimeType...
            MimeType => 'some mime type',
            Charset => 'some charset',
            TimeUnit => 123,
        },
        DynamicField => [ # optional
            {
                Name => 'some name',
                Value => 'Value', # value type depends on...
            },
            # ...
        ],
    },
};
```

(continues on next page)
Attachment => [ 
    {
        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.

In the left part of the screen on the action column you have the options to go back to web service (discarding all changes since the last save) and delete. If you click on the last one, a dialog will open and ask you if you like to remove the invoker. Click on the Delete button to confirm the removal of the invoker and its configuration or click on the Cancel button to close the delete dialog.

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.
### Invoker Details

#### General invoker data

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name is typically used to call up an operation of a remote web service.</td>
</tr>
<tr>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>Invoker backend</td>
<td>Ticket::TicketCreate</td>
</tr>
</tbody>
</table>

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 data

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ticket fields</td>
<td>Only the selected ticket fields will be considered for the request data.</td>
</tr>
<tr>
<td>Article fields</td>
<td>Only the selected article fields will be considered for the request data.</td>
</tr>
<tr>
<td>Ticket dynamic fields</td>
<td>Only the selected ticket dynamic fields will be considered for the request data.</td>
</tr>
<tr>
<td>Article dynamic fields</td>
<td>Only the selected article dynamic fields will be considered for the request data.</td>
</tr>
<tr>
<td>Number of articles</td>
<td>The outgoing request data will only contain the configured number of articles. If empty, only 1 article will be sent.</td>
</tr>
<tr>
<td>Communication channels</td>
<td>The outgoing request data will only consider articles of the selected communication channels. If empty, articles created by all communication channels will be used.</td>
</tr>
<tr>
<td>Customer visibility</td>
<td>The outgoing request data will only consider articles created with the selected customer visibility.</td>
</tr>
<tr>
<td>Sender Types</td>
<td>The outgoing request data will only consider articles created by the selected sender types. If empty, articles created by all sender types will be used.</td>
</tr>
</tbody>
</table>

#### Mapping

<table>
<thead>
<tr>
<th>Mapping for outgoing request data</th>
<th>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.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mapping for incoming response data</td>
<td>The response data will be processed by this mapping, to transform it to the kind of data the Invoker of OTRS expects.</td>
</tr>
</tbody>
</table>

#### Settings for incoming response data

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote TicketID dynamic field</td>
<td>The selected ticket dynamic field is being used to store the remote TicketID. If empty, the remote TicketID will not be stored, unless you define a system configuration value for this web service.</td>
</tr>
<tr>
<td>Ticket dynamic fields</td>
<td>Only the selected ticket dynamic fields are being considered for processing the incoming response data. If empty, no dynamic field will be processed.</td>
</tr>
</tbody>
</table>
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.

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

Normally there are two mapping configuration sections for each invoker, one for the incoming data and another one for the outgoing data. You can choose different mapping types (back ends) for each mapping direction, since their configuration is independent from each other and also independent from the invoker back end. The normal and most common practice is that the invoker uses the same mapping type in both cases, with inverted configuration. The complete mapping configuration is done in a separate screen, which depends on the mapping type.

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:

```xml
<Ticket>
  <DynamicField>..</DynamicField>
</Ticket>
```

and/or

```xml
<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 triggers are events within OTRS such as `TicketCreate`, `ArticleSend`, etc. These can act as triggers to execute the invoker. Each invoker needs to have at least one event trigger registered, or the invoker will be useless, because it will never be called. Additionally a set of rules (conditions) for each event can be defined to have more control over the triggering of the events. These rules depend on the data of the object associated with the event. The asynchronous property of the event triggers define if the OTRS process will handle the invoker or if it will be delegated to the OTRS daemon.

### Note:

The OTRS daemon is a separate set of process that executes tasks in the background. Using this the OTRS process itself will not be affected if the remote system takes a long time to respond, if it is not available or if there are network problems. If you do not use the OTRS daemon using web services can make OTRS slow or non-responsive. Therefore it is highly recommend to use asynchronous event triggers as often as possible.

In order to access this section you have to save the current invoker by clicking on the `Save` button.

### 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 plus button. Asynchronous event triggers are handled by the OTRS daemon in background (recommended). Synchronous event triggers would be processed directly during the web request.
To add an event trigger:

1. Select the event family from the first list.
2. Select the event name from the second list.
3. Set the asynchronous property. If unchecked means that the event trigger will not be asynchronous.
4. Click on the plus button. A new event trigger will be created and it will be listed on the invoker event triggers list.

From the event triggers list each events shows if it contains conditions or not. The edit button next to the condition property allows to add or edit the current conditions of the event.

To delete an event trigger, simply locate the event trigger to be deleted in the event triggers list and click on the trash icon at the end of the row. This will open a dialog that asks you if you are sure to delete the event trigger. Click on the Delete button to remove the event trigger from the list, or click on the Cancel button to close the dialog.

Sometimes defining an event to trigger an invoker could result in many unnecessary or wrong request to a remote server. Event conditions could be set to restrict the triggering of the invoker in such cases.

![General Settings](image)

**Fig. 2: Add Web Service Invoker Event Screen**

To access the event settings screen where the conditions can be defined is necessary to be in the invoker screen and from there click on the edit icon next to the condition status on the event where this condition should take effect.

Within the event settings screen in the action bar there is a button to go back to the invoker screen as well as a button to
remove all the event conditions. By default the screen is pre-populated with the first condition. Update the type of linking between conditions if more than one condition is planned, then change the type of linking from Condition 1 if more than one field is planned. Both linking fields accept and, or or xor as values.

Fill the field name, set the matching type (String for exact match, Regexp for regular expression or Validation Module) and set the value to match (in case of validation module the full class name like Kernel::GenericInterface::Event::Validation::ValidateDemo).

To add more fields to the condition, click on the plus button in the fields header. To remove a field, click on the minus button in the field row. It is necessary to keep at least one field per condition.

To add more conditions click on the button below the last condition box. To remove a condition, click on the minus button in the condition header. It is necessary to keep at least one condition in the set. To remove all conditions use the button in the sidebar.

**OTRS as Requester - HTTP::REST**

If at least two invokers are added the requester transport screen for HTTP::REST is extended with two fields.

**Controller mapping for Invoker `<InvokerName>`**

In this setting a resource path is set. This path must be defined according to the needs of the remote web service and following its definition.

Path can contain variables in the form of :<VariableName> for each variable name that matches the current data (to be sent), will be replaced by the corresponding data value. This matched variable names and values will be removed from the current data. Depending on the HTTP request command the remaining data could be sent as a JSON string in the request body or as query parameters within the URI.

Examples for data Var1 = One, Var2 = Two, Var3 = Three and Var4 = Four.

- Controller mapping: /Resource, after replacements: /Resource, remaining data: Var1 = One, Var2 = Two, Var3 = Three and Var4 = Four
- Controller mapping: /Resource/:Var1, after replacements: /Resource/One, remaining data: Var2 = Two, Var3 = Three and Var4 = Four

**Valid request command for Invoker `<InvokerName>`**

This determines the HTTP request method to use. Possible options: CONNECT, DELETE, GET, HEAD, OPTIONS, PATCH, POST, PUT and TRACE. If no command is selected, Default command is used.

### 63.2 Agent Interface

This package has no agent interface.
63.3 External Interface

This package has no external interface.
CHAPTER
SIXTYFOUR

TICKET NOTIFICATION POPUPS

Warning: This feature add-on has reached the end of life with OTRS 7 and will not be longer supported.

The functionality has been integrated into the appointment notifications and the ticket notifications of the OTRS framework.
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
- SILVER, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager
- OTRSTicketQueueSelection
65.1 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.

65.1.1 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 the 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>.

65.2 Agent Interface

This package has no agent interface.

65.3 External Interface

This package has no external interface.
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, GOLD, TITANIUM, PLATINUM

**Package Name in OTRS Package Manager**

- OTRSTicketTimeUnitDropdown
66.1 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 the 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 drop-down lists.

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.

66.2 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.

66.3 External Interface

This package has no external interface.
CHAPTER SIXTYSEVEN

TICKET WATCHLIST

This feature add-on allows you to manage several watchlists. A ticket agent defines watchlists and assigns tickets to a watchlist. Within the watchlist overview, the agent gets an overview of all of his watchlists and can administer them.

The feature add-on has the following features:

Define notification events for each watchlist

The agent is able to administer the following events which trigger an email notification to himself:

- new article
- change of customer
- change of owner
- change of queue
- change of ticket status to a defined status

This email notification is independent of the notification preferences of the current owner of the ticket – unlike the standard subscribe/watchlist feature add-on of OTRS.

Assign a watchlist to another agent

A watchlist can be handed over to another agent. The list will disappear in the overview of the former owner and appear in the overview of the new owner. This feature is typically used to hand the list over to another agent when the original agent goes on vacation.

Assign a deputy

One or more deputy agents can be assigned to a watchlist. A deputy then sees the new watchlist in his overview, and he is allowed to add or remove tickets to or from this watchlist. This feature is typically used to share a watchlist with colleagues to work on a ticket in a team.

Export to CSV

The list can be exported to a CSV file, e.g. to process it in Excel.

Benefits

- Get a quick and easy overview about all relevant tickets.
- Administration of several watchlists saves time.
- Export function enables the processing in Excel sheets.

Target Groups

- IT service management
- Service providers
- Call centers
- Service manager
Available in Service Package
- SILVER, GOLD, TITANIUM, PLATINUM

Package Name in OTRS Package Manager
- OTRSTicketWatchlist

Note: Not compatible with the following feature add-ons:
- Advanced Escalations
- Advanced Ticket Overview
- Primary / Secondary

67.1 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.

67.1.1 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.
67.2 Agent Interface

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

A watchlist can contain unlimited number of tickets and a ticket can be assigned to unlimited number of watchlists.

67.2.1 Tickets

This package provides ticket watchlists and reminders for them in the ticket detail view, as well as adds a watchlist overview.

67.2.2 Ticket Detail View

Use this screen to add tickets to a watchlist or to remove tickets from a watchlist. A new menu item Watchlists is available in the breadcrumb bar.

![Watchlists Menu](image)

Fig. 1: Watchlists Menu

The ticket can be assigned to a watchlist via this menu item by using the toggle button. Additionally, the ticket can be removed from a watchlist. The Add New Watchlist button opens the Ticket Watchlist Overview screen and the Create Ticket Watchlist action.

If the ticket is added to the watchlist, the menu item will be changed to the color defined for the watchlist. If the ticket is added to multiple watchlists, all colors will be displayed.

Use the bell icon to set a reminder for the ticket in the watchlist. For this, the agent has to be subscribed to the ticket or it has to be assigned to one of his watchlists.

**Note:** This option is visible only, if the ticket is already added to a watchlist.

The Watchlist Reminder action has the following fields by default:
Reminder active
Controls, whether the reminder is active or not. New reminders will be added only if this checkbox is checked. For existing reminders, unchecking the checkbox will remove the reminder.

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

Reminder for *
Select a watchlist from the drop-down list.

Note
Add additional information to the reminder text.

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

If a reminder is defined for the ticket in a watchlist, the bell icon will be filled.

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

In this case, where the owner of the watchlist is changed, the notification that is sent is different that the others. It can only be sent by the email notification method and it should not contain any ticket related OTRS smart tag (if so they will be removed before sending) as there is no ticket involved in the process that triggers the event.

The default Watchlist move notification is attached to the event WatchlistMoveNotification. This and any other ticket notification attached to the same event should prevent the use of these ticket related OTRS smart tags.

Reminders will only be sent if the assigned agent at this time still has access to the watchlist containing the ticket and/or still is subscribed to the ticket.
67.2.3 Ticket Watchlist Overview

Use this screen to manage watchlists. The ticket watchlists overview screen is available from the organizer via the glasses icon.

This screen is not designed for accessing tickets which are assigned to a ticket watchlist. To see all tickets which are assigned to a ticket watchlist, there is a new filter Watchlist available in all ticket lists.

Manage Watchlists

To add a watchlist:

1. Click on the plus icon in the breadcrumb bar.
2. Fill in the required field.
3. Click on the Save button.

To export the ticket list from a watchlist:

1. Click on the export button in the Actions column of a watchlist.
2. Select the export format.
3. Click on the Download File button.
4. Save the file somewhere in the file system.

To edit a watchlist:

1. Click on the edit button in the Actions column of a watchlist.
2. Modify the field.
3. Click on the Save button.
To change the watchlist owner:

1. Click on the person icon in the *Actions* column of a watchlist.
2. Select an agent from the list.
3. Click on the *Save* button.

**Warning:** The original owner agent will not have access to the watchlist anymore.

To assign deputies to a watchlist:

1. Click on the police badge icon in the *Actions* column of a watchlist.
2. Select deputy agents from the list.
3. Click on the *Save* button.

To delete a watchlist:

1. Click on the trash icon in the *Actions* column of a watchlist.
2. Click on the *Yes* button in the confirmation dialog.

Tickets can be added to the watchlist or removed from the watchlist in the *Tickets* screen.

**Watchlist Settings**

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

**Name * **

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

**Color * **

The color associated to the watchlist. This color will be displayed in the ticket detail view when a watchlist is selected for the ticket.

**Notify On**

Select the events that will trigger a notification.

- **New Agent Article**
  Select, if the watchlist owner should be notified, when an agent adds a note to a ticket added to the watchlist.

- **New Customer Article**
  Select, if the watchlist owner should be notified, when a customer user creates an article in the external interface.

- **New Owner**
  Select, if the watchlist owner should be notified, when a ticket added to the watchlist has a new owner.

- **New Queue**
  Select, if the watchlist owner should be notified, when a ticket added to the watchlist has been moved to another queue.

- **New State**
  Select, if the watchlist owner should be notified, when a state is changed for a ticket added to the watchlist.
Target States

Notifications for status changes are only being sent if the new status matches the configured destination status. If the *New State* notification above is selected, this field becomes mandatory.

Watchlist owners and their deputies can configure the notifications separately. The configuration of the destination status is only allowed for the watchlist owner and therefore also affects the deputies.

Notifications are not being sent to the agent who triggered the action.

See also:
The content of the notifications can be edited in the ticket detail view when the ticket is assigned to a watchlist. See the *Tickets* chapter for more details.

67.3 External Interface

This package has no external interface.
Warning: This feature add-on has reached the end of life with OTRS 7 and will not be longer supported.

The functionality has been integrated into the process management of the OTRS framework, the Configuration Management and the Calendar Resource Planning.
Warning: This feature add-on has reached the end of life with OTRS 7 and will not be longer supported.

You can use the calendar feature to manage projects and tasks. In the calendar you can assign tasks to different groups and agents and you can use a number of other possibilities, such as automation and linking tasks to different objects.

Note: Before the migration to OTRS you have to save all data. After the migration the data will not be available.
This feature add-on is ideal for call centers, IT service providers or any company that wants to assign time or quantity-based quotas to their customers.

The feature add-on provides a dashboard widget in the detail view of the respective customer that displays time unit quotas or volume quotas with reference numbers to tickets and the current status of the quota. The time unit and volume quotas are maintained in the CMDB and they can be assigned to the respective customers.

As an alternative to calculating times, the amount of incident reports, for example, can also be taken as the basis for calculating the number still available.

**Benefits**

- Simple time recording and transparent documentation in one system.
- Having an overview of current quotas facilitates planning and implementation.

**Target Groups**

- IT service management
- Call centers
- Service providers
- All companies

**Available in Service Package**

- SILVER, GOLD, TITANIUM, PLATINUM

**Package Name in OTRS Package Manager**

- OTRSTimeAndQuotaManagement

---

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

This feature add-on requires the *Ticket Time Unit Dropdown* feature.
70.1 Administrator Interface

This module adds additional functionality to set time contingents for customers.

70.1.1 Administration

After installation of the package some new classes will be available in the General Catalog.

General Catalog

This package adds some new classes to the General Catalog. The general catalog management screen is available in the General Catalog module of the Administration group.

<table>
<thead>
<tr>
<th>List</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATALOG CLASS</td>
</tr>
<tr>
<td>ITSM::ConfigItem::Class</td>
</tr>
<tr>
<td>ITSM::ConfigItem::Computer::Type</td>
</tr>
<tr>
<td>ITSM::ConfigItem::DeploymentState</td>
</tr>
<tr>
<td>ITSM::ConfigItem::Hardware::Type</td>
</tr>
<tr>
<td>ITSM::ConfigItem::Location::Type</td>
</tr>
<tr>
<td>ITSM::ConfigItem::Network::Type</td>
</tr>
<tr>
<td>ITSM::ConfigItem::QuotaManagement::NotifyBy</td>
</tr>
<tr>
<td>ITSM::ConfigItem::QuotaManagement::RenewalFrequency</td>
</tr>
<tr>
<td>ITSM::ConfigItem::Software::LicenceType</td>
</tr>
<tr>
<td>ITSM::ConfigItem::Software::Type</td>
</tr>
<tr>
<td>ITSM::ConfigItem::YesNo</td>
</tr>
<tr>
<td>ITSM::Core::IncidentState</td>
</tr>
<tr>
<td>ITSM::Service::Type</td>
</tr>
<tr>
<td>ITSM::SLA::Type</td>
</tr>
</tbody>
</table>

Fig. 1: General Catalog Class List Screen

New Classes

ITSM::ConfigItem::Class

A class for configuration item classes.

Two new classes are added to this class to store the quotas:

- Time Unit Quota
- Volume Quota

See also:

The class definition of configuration item classes can be managed in the Configuration Items module of the CMDB Settings group.

ITSM::ConfigItem::QuotaManagement::NotifyBy

A class for specifying the percentage of the quota when notification should be sent.
ITSM::ConfigItem::QuotaManagement::RenewalFrequency
A class for specifying the renewal frequency of the quota.

<table>
<thead>
<tr>
<th>Catalog Class:</th>
<th>ITSM::ConfigItem::QuotaManagement::RenewalFrequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name:</td>
<td>Each Month</td>
</tr>
<tr>
<td>Renew Interval:</td>
<td>1</td>
</tr>
<tr>
<td>Renew Time Unit:</td>
<td>Month</td>
</tr>
<tr>
<td>Validity:</td>
<td>valid</td>
</tr>
<tr>
<td>Comment:</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 2: Quota Management Renewal Frequency

70.2 Agent Interface

This package extends the quota functionality to be able to set different quota for each customer.

70.2.1 Configuration Items

Quotas will be managed in specific configuration item classes in the General Catalog. This simplifies the handling of quotas. There is versioning of quota configuration items so that the latest version of the quota is always up-to-date and the quota history remains audit-proof.

The quotas are now divided into time-based and volume-based quotas. Time-based quotas are reduced by time units accounted in each article while volume-based quotas are reduced by 1 per tickets. This allows our customers to choose the mode in which they offer their contracts.

Since time-based and volume-based quotas have major differences in their specific handling, there are two separate system configuration settings specifically designed for each use case. Quotas can be added like any other configuration item. The class definition is the same for both quota types.

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

Customer Company *
The customer where the quota is assigned to.

Quota Total *
Total amount of quota. This amount will be renewed if needed.

Used Quota *
How much quota is used by the customer at the creation time of the configuration item. Set to 0 if no quota is used yet.

Remaining Quota *
How much quota is remaining for the customer. Set the same amount as Quota Total if no quota is used yet.
Fig. 3: Quota Properties
Last Used Reference
Reference to the ticket where the quota is used. Leave it empty if no quota is used yet.

Notification Percentage
A notification will be sent if the quota usage reached this percentage.

Notification Sent *
Whether a notification was sent. Set to No if no quota is used yet.

Automatic Renewal
Quota renewal possibility after the configured renewal frequency.

Renewal Repetition Limit
How many times should the quota be renewed.

Renewal Time Limit
What is the last date for the quota renewal.

Last Renewal
The date of the last quota renewal.

Renewal Count
How many renewal happened.

The newly created configuration item will be displayed in the Configuration Item Overview screen and in the Customers detail view of the relevant customer. If any renewal option has been set, the quota will be reset after the configured time period.

### 70.2.2 Customers

The quota lists are displayed in the customer detail view as separate widgets.

The Time Unit Quota List widget displays the total quota, the used quota and the remaining quota. The Last Used Reference column displays the ticket number and article ID that is used for changing the quota in format TicketNumber#ArticleID.

<table>
<thead>
<tr>
<th>Time Unit Quota List (1)</th>
<th>Select Preset</th>
<th>Search</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Quota</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>20.00</td>
<td>80.00</td>
</tr>
</tbody>
</table>

Fig. 4: Time Unit Quota List

In the example above the customer has 100 units quota. A ticket was created for a customer user who belongs to this customer, and 20 time units was accounted. The value 2022052610000016#43 in the Last Used Reference column means that ticket number 2022052610000016 and article ID 43 changed the quota last.

To change the time unit quota of a customer:

1. Create a new ticket for the customer or add an article to an existing ticket of the customer.
2. Add any number to the Time Units field.
3. Go to the customer detail view and verify that the quota is reduced by the time unit added to the article.

The Volume Quota List widget displays the total quota, the used quota and the remaining quota. The Last Used Reference column displays the ticket number that is used for changing the quota.

In the example above the customer has 5 service requests quota. A ticket was created for a customer user who belongs to this customer, and the quota was reduced by 1. The value 2022052610000025 in the Last Used Reference column means that ticket number 2022052610000025 changed the quota last.

70.2. Agent Interface
To change the volume quota of a customer:

1. Create a new ticket for the customer.
2. Go to the customer detail view and verify that the quota is reduced by 1.

### 70.3 External Interface

This package has no external interface.
Warning: This feature add-on has reached the end of life with OTRS 7 and will not be longer supported.

Note: If this feature add-on is already installed, the feature will be replaced by the core functionality of the framework during the upgrading process. After the upgrading process, the package will be removed.