

## **Documentation**

# **OTRS Business Solution™ 4 Manual**

**Build Date:**

**2014-12-05**

---

---

# OTRS Business Solution™ 4 Manual

Copyright © 2014 OTRS AG

This work is copyrighted by OTRS AG. You may copy it in whole or in part as long as the copies retain this copyright statement.

UNIX is a registered trademark of X/Open Company Limited. Linux is a registered trademark of Linus Torvalds. MS-DOS, Windows, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows 2003, Windows Vista and Windows 7 are registered trademarks of Microsoft Corporation. Other trademarks and registered trademarks are: SUSE and YaST of SUSE Linux GmbH, Red Hat and Fedora are registered trademarks of Red Hat, Inc. Mandrake is a registered trademark of MandrakeSoft, SA. Debian is a registered trademark of Software in the Public Interest, Inc. MySQL and the MySQL Logo are registered trademarks of Oracle Corporation and/or its affiliates. All trade names are used without the guarantee for their free use and are possibly registered trade marks. OTRS AG essentially follows the notations of the manufacturers. Other products mentioned in this manual may be trademarks of the respective manufacturer.



# Table of Contents

1. Introduction .....	1
2. Upgrading to <b>OTRS Business Solution</b> ™ .....	2
3. Features of <b>OTRS Business Solution</b> ™ .....	3
1. The Ticket Timeline View .....	3
1.1. Description .....	3
1.2. Configuration .....	3
1.3. Usage .....	3
2. The Article Attachment Overview .....	4
2.1. Description .....	4
2.2. Configuration .....	4
2.3. Usage .....	4
3. The Chat .....	5
3.1. Description .....	5
3.2. Configuration .....	5
3.3. Usage .....	6
4. Field Selection Dialog .....	8
4.1. Description .....	8
4.2. Configuration .....	8
4.3. Usage .....	9
5. Contact with Data .....	10
5.1. Description .....	10
5.2. Configuration .....	11
5.3. Usage .....	11
6. Dynamic Field Database .....	12
6.1. Description .....	12
6.2. Configuration .....	12
6.3. Usage .....	16

---

## List of Figures

3.1. Switching to the ticket timeline view .....	3
3.2. Using the ticket timeline view .....	4
3.3. Switching to the article attachment overview .....	4
3.4. Using the article attachment overview .....	5
3.5. The chat managing screen .....	6
3.6. The chat widget .....	7
3.7. Start a chat .....	7
3.8. The chat actions .....	8
3.9. SLA configuration .....	9
3.10. SLA message .....	10
3.11. DynamicField Database Admin screen .....	13
3.12. DynamicField Database - Historical data settings .....	15
3.13. DynamicField Database - Test field .....	16
3.14. DynamicField Database - Autocomplete feature .....	17
3.15. DynamicField Database - Selected items .....	17
3.16. DynamicField Database - Detailed search .....	17
3.17. DynamicField Database - Detailed search result .....	18



---

# Chapter 1. Introduction

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

The sleek and fast framework of OTRS 4 is the base for the **OTRS Business Solution™**. On top are available for you additionally an exclusive set of features. These include the **integrated Cloud Services** of the **OTRS Business Solution™** that you can flexibly activate or deactivate according to your deployment scenarios. Additionally you have the choice between 3, 7 or an unlimited number of various **freely selectable OTRS Business Solution™ Features** - depending on you contract level. Profit also from regular, free updates, the best OTRS security and support as well as installation and configuration carried out by our consultants.



---

# Chapter 2. Upgrading to OTRS Business Solution™

In order to upgrade to **OTRS Business Solution™**, you need a valid contract with the OTRS group. If you already have a contract, you can upgrade to **OTRS Business Solution™** by visiting the AdminOTRSBusiness screen in your OTRS instance. If you're entitled to upgrade, you will (as an admin user) also see a notification bar which guides you to the upgrade screen. The upgrade itself is done automatically by just pressing the upgrade button.

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



# Chapter 3. Features of OTRS Business Solution™

## 1. The Ticket Timeline View

### 1.1. Description

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

### 1.2. Configuration

#### 1.2.1. ChronicalViewEnabled

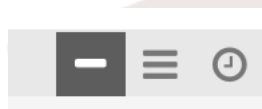
Group: OTRSBusiness, Subgroup: Frontend::Agent

Controls whether or not the ticket timeline view is enabled.

### 1.3. Usage

#### 1.3.1. Accessing the ticket timeline view

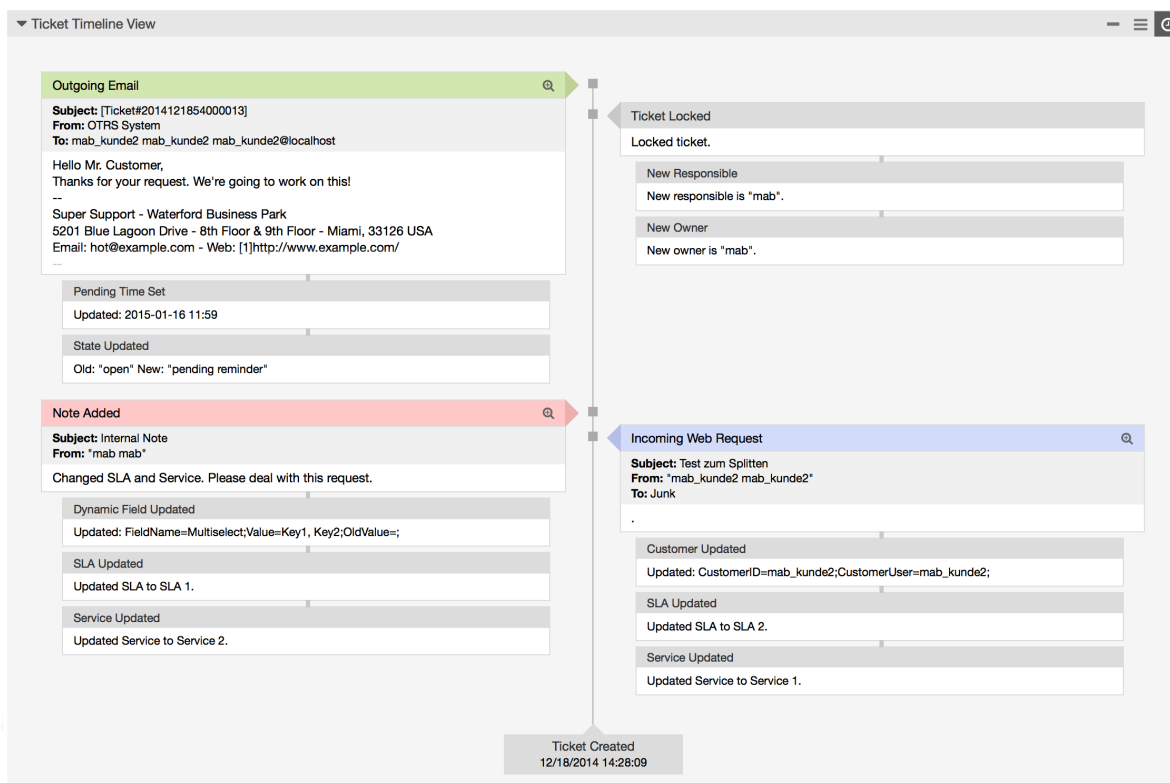
**Figure 3.1. Switching to the ticket timeline view**



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

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

**Figure 3.2. Using the ticket timeline view**



## 2. The Article Attachment Overview

### 2.1. Description

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

### 2.2. Configuration

No further configuration needed.

### 2.3. Usage

#### 2.3.1. Accessing the article attachment overview

**Figure 3.3. Switching to the article attachment overview**

nk | Owner | Responsible | Attachments | Customer | Note | PI

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



**Figure 3.4. Using the article attachment overview**

**Attachments of Ticket#: 201412185400013 - Test zum Splitten**  
 Close window

Attachment Overview

Filter Attachments...

<input type="checkbox"/>	TYPE	FILENAME	SIZE	▼ DATE	≡	ARTICLE
<input type="checkbox"/>		chat-manager.jpeg	152.2 KBytes	01/15/2015 12:06	<input type="checkbox"/>	#4 - Note
<input type="checkbox"/>		cover1.jpg	81.9 KBytes	01/15/2015 12:06	<input type="checkbox"/>	#4 - Note
<input type="checkbox"/>		vector_dancers_by_blindblues46.pdf	512.4 KBytes	01/15/2015 12:06	<input type="checkbox"/>	#4 - Note

Close

## 3. The Chat

### 3.1. Description

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

### 3.2. Configuration

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

#### 3.2.1. ChatEngine::Active

Group: OTRSBusiness, Subgroup: Core::Chat

Controls whether or not the chat feature is enabled.

#### 3.2.2. ChatEngine::ChatDirection::PublicToAgent

Group: OTRSBusiness, Subgroup: Core::Chat

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

#### 3.2.3. ChatEngine::ChatDirection::CustomerToAgent

Group: OTRSBusiness, Subgroup: Core::Chat

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

#### 3.2.4. ChatEngine::ChatDirection::AgentToCustomer

Group: OTRSBusiness, Subgroup: Core::Chat

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

### 3.2.5. ChatEngine::ChatDirection::AgentToAgent

Group: OTRSBusiness, Subgroup: Core::Chat

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

### 3.2.6. ChatEngine::PermissionGroup::ChatReceiv- ingAgents, ChatEngine::PermissionGroup::ChatNotifi- cationAgents, ChatEngine::PermissionGroup::ChatS- tartingAgents

Group: OTRSBusiness, Subgroup: Core::Chat

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

### 3.2.7. ChatEngine::DefaultAgentName

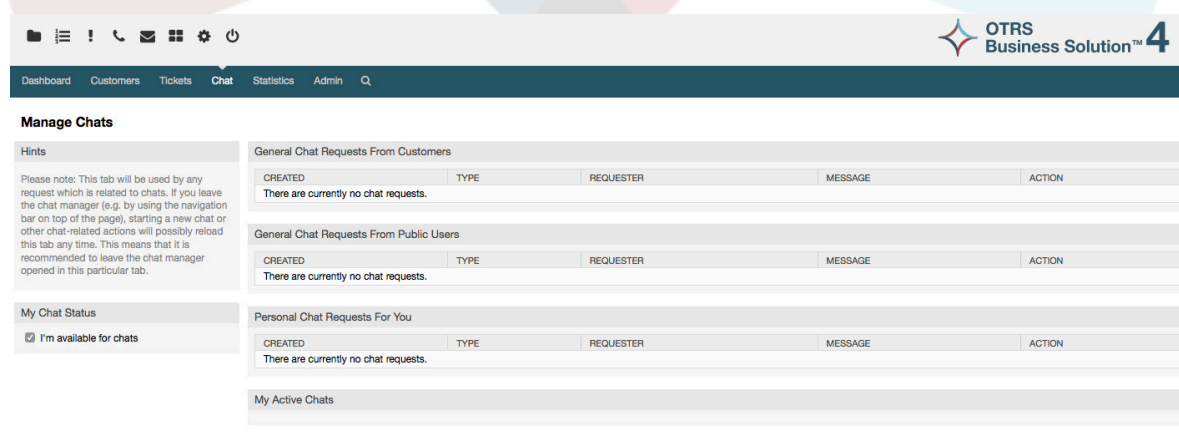
Group: OTRSBusiness, Subgroup: Core::Chat

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

## 3.3. Usage

### 3.3.1. The chat manager

**Figure 3.5. The chat managing screen**

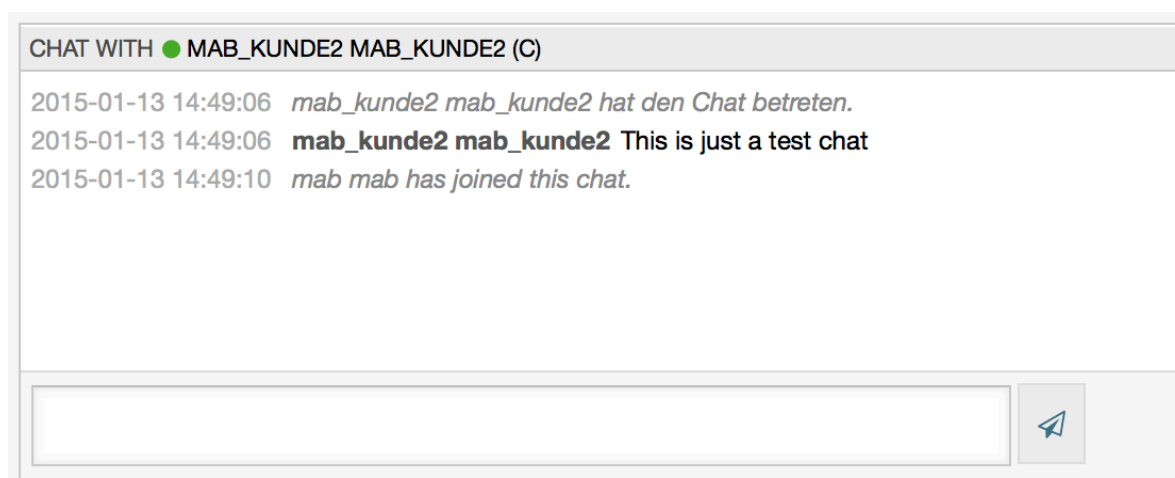


The chat manager is the central point of the chat feature. As an agent, here you can manage your chats, see and accept open requests and change your chat availability status. The chat manager provides a list of chat requests from customers, public users and other agents, as well as an overview over all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

You can set your general availability for the chat by using the checkbox in the left sidebar of the chat manager. Once disabled, you'll no longer get notified for incoming chat requests.

## 3.3.2. Using the chat widget

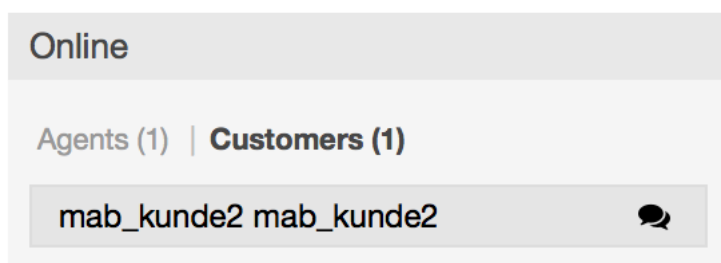
**Figure 3.6. The chat widget**



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

## 3.3.3. Starting new chats and dealing with chat requests

**Figure 3.7. Start a chat**



As an agent, there are multiple ways to start a chat. First of all, you can use the user online widget on the dashboard to chat with agents or customers who are currently logged in to OTRS. Just click the little chat icon behind the name of the agent or customer. A small overlay will appear and allow for entering the first chat message you would like to send to the other user. Afterwards, the new chat is being added to the chat manager.

You can also use ticket zoom to start a chat with a certain customer. In the sidebar, you'll find a chat link in the customer information widget.

If new chat requests arrive and you're logged in to OTRS, you'll get notified by a chat icon in the toolbar on top of the screen. The icon also holds information about how many open chat requests are available at the moment. Just click the icon to open the chat manager. You'll also receive a browser notification (popup) in case of new chat requests.

All agents with the needed permissions will be able to deal with open chat requests. A request can only be accepted by one agent at the same time, which means that the request will disappear from the other agents lists once it has been accepted by an agent.

Once accepted, the newly created chat will be appended to the list of active chats for the agent who has accepted it. Other users in the chat will see a notification about the user who just entered the chat.

### 3.3.4. Ending and archiving chats

**Figure 3.8. The chat actions**



As an agent, you have several possibilities to end a chat. First of all, you can just discard a chat by using the "x" icon on the right top side of the chat widget in the chat manager. The chat then just will be closed without archiving it. Other users in the chat will just see a message telling them the chat was ended.

You can also append the chat to an existing ticket. Click the "+" icon on the right top of the chat widget in the chat manager. A new window will open which allows for selecting a ticket which the chat should be appended to.

If you don't have any appropriate ticket to append the chat to, you can also create a new one by using the phone icon on the top right side of the chat widget in the chat manager.

## 4. Field Selection Dialog

### 4.1. Description

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

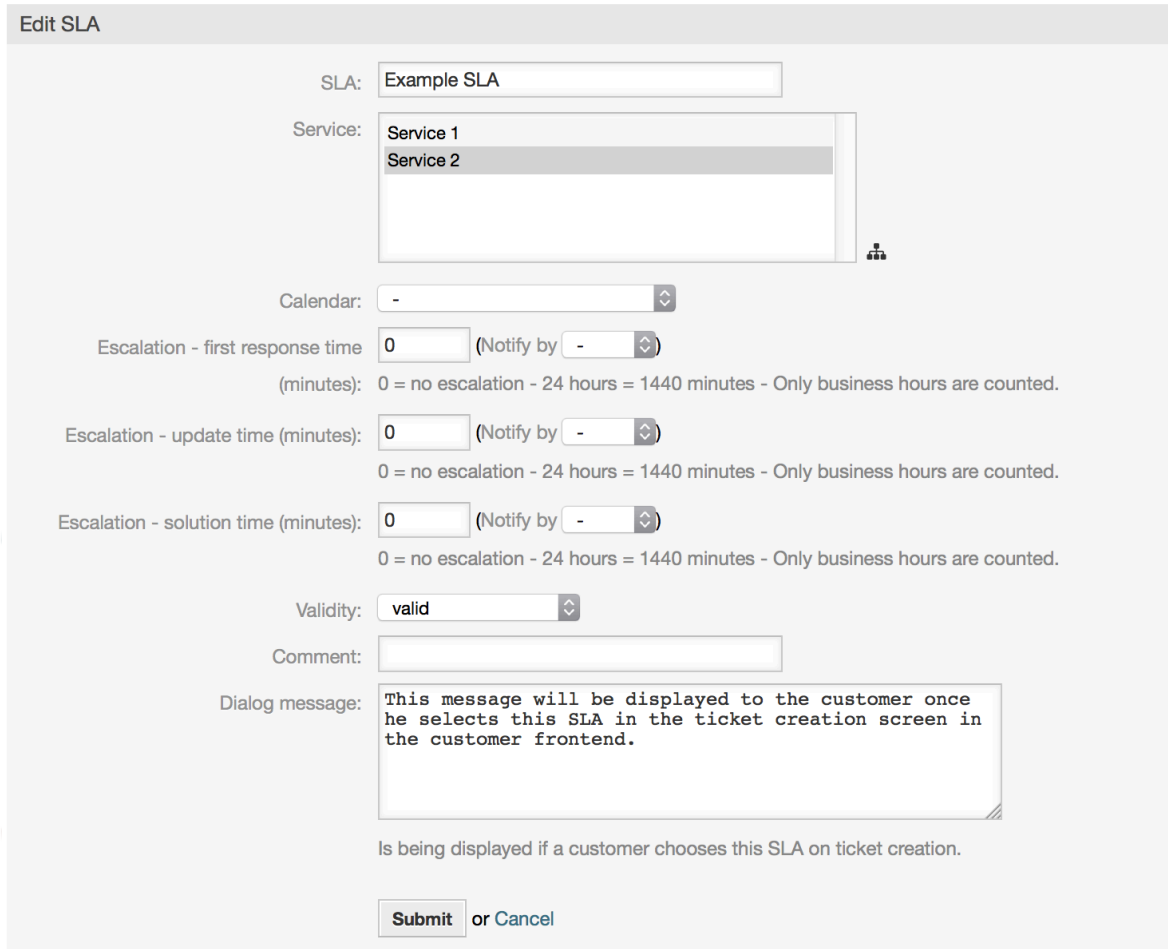
### 4.2. Configuration

No further configuration needed.

## 4.3. Usage

### 4.3.1. Configuring the SLA message

**Figure 3.9. SLA configuration**



**Edit SLA**

SLA:

Service: 

- Service 1
- Service 2**

Calendar:

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

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

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

Validity:

Comment:

Dialog message: 

This message will be displayed to the customer once he selects this SLA in the ticket creation screen in the customer frontend.

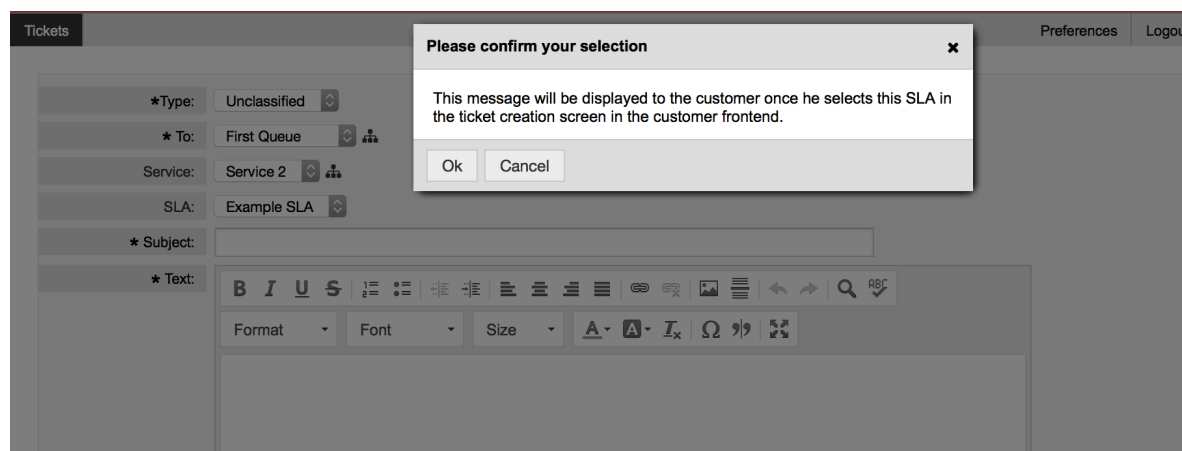
Is being displayed if a customer chooses this SLA on ticket creation.

or

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

## 4.3.2. How the message looks in the customer frontend

**Figure 3.10. SLA message**



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

## 5. Contact with Data

This feature allows to add contacts with data to tickets.

### 5.1. Description

#### 5.1.1. Definition of data sources

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

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

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

#### Note

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

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

#### 5.1.2. Contact management

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

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

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

### 5.1.3. Adding contacts to tickets

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

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

### 5.1.4. Usage of contacts in search and statistics

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

The 'Name' can also be returned as a field value.

## 5.2. Configuration

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

### 5.2.1. AdminContactWithData::RunInitialWildcardSearch

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

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

### 5.2.2. Frontend::Module###AdminContactWithData

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

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

## 5.3. Usage

An exemplary usage of contacts with data is as follows:

1. Create a new dynamic field of type 'Contact with data'.
2. Set the possible contact attributes (possible values). 'Name' and 'ValidID' are required for any Contact with data Dynamic Field.
  - Add 'Name' attribute (Key: Name, Value: 'Name').
  - Add 'ValidID' attribute (Key: ValidID, Value: 'Validity').
  - Add any other attribute such as 'Telephone' attribute (Key: Telephone, Value: 'Phone').
3. Add the list of mandatory attribute keys comma separated ('Name' and 'ValidID' are not needed).

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

## 6. Dynamic Field Database

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

### 6.1. Description

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

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

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

### 6.2. Configuration

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

#### 6.2.1. Add a dynamic field 'Database'

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

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



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

Configuration "General - Label": The label can be individually set and can contain white spaces etc. It will be used as the field label in the different views.

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

Configuration "General - Valid": To use the dynamic field in the configured OTRS 'views' it has to be set as valid. If the field is set to 'invalid' it will disappear from all the configured views but no data will be lost.

## 6.2.2. Configuration of the external source (Database)

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

**Figure 3.11. DynamicField Database Admin screen**

Dynamic Fields - Ticket: Change Database Field

Actions

[Go back to overview](#)

General

\* Name:  Validity:

Must be unique and only accept alphabetic and numeric characters.

\* Label:  Field type:

This is the name to be shown on the screens where the field is active.

\* Field order:  Object type:

This is the order in which this field will be shown on the screens where is active.

---

Database Field Settings

Possible values:

- \* Name:  \* Label:  \* Datatype:  Filter:  Searchfield:  Listfield:
- \* Name:  \* Label:  \* Datatype:  Filter:  Searchfield:  Listfield:
- \* Name:  \* Label:  \* Datatype:  Filter:  Searchfield:  Listfield:
- \* Name:  \* Label:  \* Datatype:  Filter:  Searchfield:  Listfield:

Add value:

Show link:

Here you can specify an optional HTTP link for the field value in Overviews and Zoom screens.  
Example: [http://some.example.com/handle?query=\\${QData\['Field1'\]}](http://some.example.com/handle?query=${QData['Field1']})

\* Type:

\* Server:

Port:

\* Database:

\* Table / View:

\* User:

\* Password:

\* Identifier:

Multiselect:

CacheTTL:

Searchprefix:

Searchsuffix:

Result Limit:

Case Sensitive:

or

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

SID: This option is only available for Oracle connections and will be shown or hidden automatically. Within this option you have to enter the SID of your Oracle connection.

Driver: This option is only available for ODBC connections and will be shown or hidden automatically. Within this option you have to enter in the hostsystem previously configured ODBC driver to connect to the desired MSSQL database.

---

**Server:** The database host (hostname or IP-Address).

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

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

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

**User:** The username for the database connection.

**Password:** The user password for the database connection.

**Identifier:** This select box will be automatically filled through "Possible Values (description below)". This field represents the value which will be stored in the dynamic field.

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

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

**Searchprefix:** This value will be put in the front of every search term while using the autocompletion to search the database. Wildcard characters are supported as well. The searchprefix will be ignored during the detailed search, but it is still possible to use wildcard characters in those masks.

**Searchsuffix:** This value will be put in the end of every search term while using the autocompletion to search the database. Wildcard characters are supported as well. The searchsuffix will be ignored during the detailed search, but it is still possible to use wildcard characters in those masks.

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

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

**Possible values:** As already explained the possible values will fill up the identifier field automatically, which defines the value that will be stored in the dynamic field. Possible values can be created as much as needed (or at least as many table columns as the database table has). The possible values defines the database columns to search in. It is possible to set the column name, a description (label) the field should have, the needed data type and if the field should be a search- or listfield.

**Name:** The exact name of the database column which will be requested through the database queries.

**Description:** The label of the field which will be displayed in the detailed search.

**Datatype:** The data type which will be stored in the dynamic field. Possible values: TEXT, INTEGER or DATE.

**Filter:** With the filter field, it is possible to choose a ticket attribute or a dynamic field as a filter for the related column. If the dynamic field is bound to a related ticket, the attributes will be used for the filter mechanism, otherwise the filters will be ignored. If filter will be configured to a table column, only search results matching to the search term and the related ticket attribute on exactly the configured column will be displayed.

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

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

### 6.2.3. Configure dynamic fields of type 'Database' to be displayed in the agent interface

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

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

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

Frontend::Agent::Ticket::ViewZoom for the ticket zoom view.

Frontend::Agent::Ticket::ViewPhoneNew for new phone tickets.

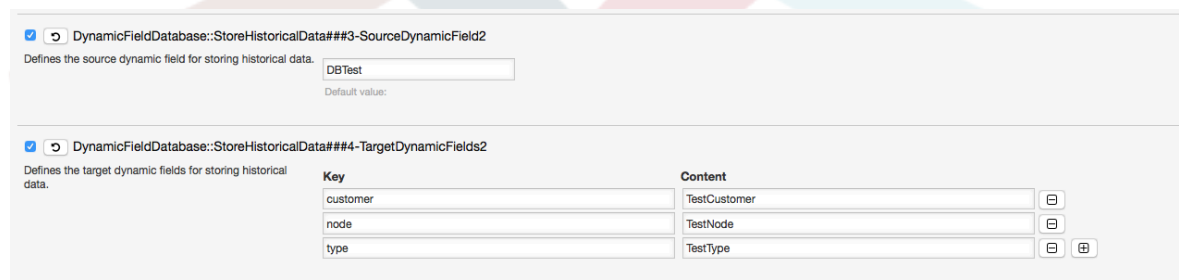
Frontend::Agent::Ticket::ViewEmailNew for new email tickets.

In each of these view configurations is an entry called "Ticket::Frontend::AgentTicket\*###DynamicField". This configuration defines which dynamic field(s) should get displayed in this view. To add a dynamic field, the internal name of the field has to be filled in the 'Key' field. The field 'Value' can take the values 0 (deactivated), 1 (active) and 2 (active and mandatory).

### 6.2.4. Storage of historical data

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

**Figure 3.12. DynamicField Database - Historical data settings**



The screenshot shows the SysConfig interface for 'DynamicField Database - Historical data settings'. It contains two main sections:

- DynamicFieldDatabase::StoreHistoricalData###3-SourceDynamicField2**: This section is checked and contains a text input field for 'SourceDynamicField2' with the value 'DBTest' and a 'Default value:' label below it.
- DynamicFieldDatabase::StoreHistoricalData###4-TargetDynamicFields2**: This section is also checked and contains a table for defining target dynamic fields. The table has two columns: 'Key' and 'Content'.
 

Key	Content
customer	TestCustomer
node	TestNode
type	TestType

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

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

### 6.2.5. Sysconfig settings

#### 6.2.5.1. AutoComplete::Agent###DynamicFieldDatabaseSearch

Group: Framework, Subgroup: Frontend::Agent

Defines the config options for the autocompletion feature.

### **6.2.5.2. Ticket::EventModulePost###950-StoreHistoricalData**

Group: Ticket, Subgroup: Core::Ticket

Updates dynamic fields, if configured ones will be updated.

### **6.2.5.3. DynamicFieldDatabase::StoreHistoricalData###1-Source-DynamicField1**

Group: OTRSBusiness, Subgroup: Core

Defines the source dynamic field for storing historical data.

### **6.2.5.4. DynamicFieldDatabase::StoreHistoricalData###2-Target-DynamicFields1**

Group: OTRSBusiness, Subgroup: Core

Defines the target dynamic fields for storing historical data.

### **6.2.5.5. DynamicFieldDatabase::StoreHistoricalData###3-Source-DynamicField2**

Group: OTRSBusiness, Subgroup: Core

Defines the source dynamic field for storing historical data.

### **6.2.5.6. DynamicFieldDatabase::StoreHistoricalData###4-Target-DynamicFields2**

Group: OTRSBusiness, Subgroup: Core

Defines the target dynamic fields for storing historical data.

## **6.3. Usage**

An exemplary usage of DynamicField Database is as follows:

### **6.3.1. Searching and saving datasets - Autocompletion**


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

#### **Figure 3.13. DynamicField Database - Test field**

DB Test:  

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

### Figure 3.14. DynamicField Database - Autocomplete feature

DB Test:  




(work units):

NodeOne - CustomerOne - TypeOne
NodeTwo - CustomerTwo - TypeTwo

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

### Figure 3.15. DynamicField Database - Selected items

DB Test:

NodeOne - CustomerOne - TypeOne	 
NodeTwo - CustomerTwo - TypeTwo	 

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.


## 6.3.2. Searching and saving datasets - Detailed search


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


### Figure 3.16. DynamicField Database - Detailed search


**Detailed search** x

ID:  

Node:  

Customer:  

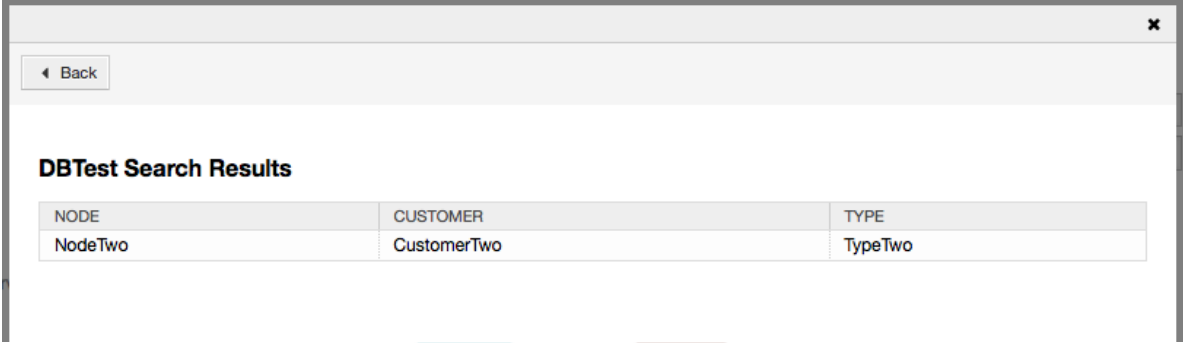
Type:  

Add an additional attribute:   

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

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

**Figure 3.17. DynamicField Database - Detailed search result**



NODE	CUSTOMER	TYPE
NodeTwo	CustomerTwo	TypeTwo

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