

# **Documentation**

## **OTRSMasterSlave**

**Includes "Ticket Master/Slave" feature. Version 6.0.15 Edition**

**Build Date:**

**2019-05-18**

---

# Table of Contents

Preface .....	iv
1. Feature List .....	1
1. Basic Mode .....	1
2. Advanced Mode .....	1
2.1. MasterSlave Screen .....	1
2.2. Bulk Action .....	2
3. Replace customer's from Master ticket .....	2
2. System Requirements .....	3
1. Framework .....	3
2. Packages .....	3
3. Operating System .....	3
4. Third Party Software .....	3
3. Installation .....	4
1. Admin Interface .....	4
2. Command Line .....	4
4. Configuration .....	5
1. ACLKeysLevel3::Actions###888-OTRSMasterSlave. ....	5
2. DashboardBackend###0900-TicketMaster. ....	5
3. DashboardBackend###0910-TicketSlave. ....	5
4. DynamicFields::Driver###MasterSlave. ....	5
5. Frontend::Module###AdminDynamicFieldMasterSlave. ....	5
6. Frontend::Module###AgentTicketMasterSlave. ....	5
7. Frontend::Navigation###AdminDynamicFieldMasterSlave###004-OTRSMas-	
terSlave. ....	5
8. Frontend::Navigation###AgentTicketMasterSlave###004-OTRSMasterSlave. ....	5
9. Loader::Module::AdminDynamicFieldMasterSlave###004-OTRSMasterSlave. ....	6
10. Loader::Module::AgentTicketMasterSlave###004-OTRSMasterSlave. ....	6
11. MasterSlave::AdvancedEnabled. ....	6
12. MasterSlave::DynamicField. ....	6
13. MasterSlave::FollowUpdatedMaster. ....	6
14. MasterSlave::ForwardSlaves. ....	6
15. MasterSlave::KeepParentChildAfterUnset. ....	6
16. MasterSlave::KeepParentChildAfterUpdate. ....	6
17. MasterSlave::UnsetMasterSlave. ....	6
18. MasterSlave::UpdateMasterSlave. ....	7
19. PreApplicationModule###AgentPreMasterSlave. ....	7
20. ReplaceCustomerRealNameOnSlaveArticleTypes. ....	7
21. ReplaceCustomerRealNameOnSlaveArticleCommunicationChannels. ....	7
22. Ticket::EventModulePost###MasterSlave. ....	7
23. Ticket::Frontend::AgentTicketMasterSlave###ArticleTypeDefault. ....	7
24. Ticket::Frontend::AgentTicketMasterSlave###ArticleTypes. ....	7
25. Ticket::Frontend::AgentTicketMasterSlave###Body. ....	7
26. Ticket::Frontend::AgentTicketMasterSlave###HistoryComment. ....	7
27. Ticket::Frontend::AgentTicketMasterSlave###HistoryType. ....	8
28. Ticket::Frontend::AgentTicketMasterSlave###InformAgent. ....	8
29. Ticket::Frontend::AgentTicketMasterSlave###InvolvedAgent. ....	8
30. Ticket::Frontend::AgentTicketMasterSlave###MasterSlaveMandatory. ....	8
31. Ticket::Frontend::AgentTicketMasterSlave###Note. ....	8
32. Ticket::Frontend::AgentTicketMasterSlave###NoteMandatory. ....	8
33. Ticket::Frontend::AgentTicketMasterSlave###Owner. ....	8
34. Ticket::Frontend::AgentTicketMasterSlave###OwnerMandatory. ....	8

35. Ticket::Frontend::AgentTicketMasterSlave###Permission. ....	8
36. Ticket::Frontend::AgentTicketMasterSlave###Priority. ....	9
37. Ticket::Frontend::AgentTicketMasterSlave###PriorityDefault. ....	9
38. Ticket::Frontend::AgentTicketMasterSlave###RequiredLock. ....	9
39. Ticket::Frontend::AgentTicketMasterSlave###Responsible. ....	9
40. Ticket::Frontend::AgentTicketMasterSlave###ResponsibleMandatory. ....	9
41. Ticket::Frontend::AgentTicketMasterSlave###RichTextHeight. ....	9
42. Ticket::Frontend::AgentTicketMasterSlave###RichTextWidth. ....	9
43. Ticket::Frontend::AgentTicketMasterSlave###Service. ....	9
44. Ticket::Frontend::AgentTicketMasterSlave###State. ....	10
45. Ticket::Frontend::AgentTicketMasterSlave###StateDefault. ....	10
46. Ticket::Frontend::AgentTicketMasterSlave###StateType. ....	10
47. Ticket::Frontend::AgentTicketMasterSlave###Subject. ....	10
48. Ticket::Frontend::AgentTicketMasterSlave###TicketType. ....	10
49. Ticket::Frontend::AgentTicketMasterSlave###Title. ....	10
50. Ticket::Frontend::BulkModule###010-MasterSlave. ....	10
51. Ticket::Frontend::MenuModule###480-MasterSlave. ....	10
5. Usage .....	11
1. Create a Master Ticket .....	11
2. Create A Slave Ticket .....	11
3. Make Slave ticket To Follow Master Updates .....	11
4. Change The Master Slave Status .....	12
6. Technical Implementation Details .....	13
7. File list .....	14
8. Tests .....	16
1. Test Cases .....	16
2. Unit Tests .....	16
9. Change Log .....	17

---

## Preface

This package adds the support to create slaves tickets from a master that follows their master updates.

If you have questions regarding this document or if you need further information, please log in to our customer portal at [portal.otrs.com](https://portal.otrs.com) with your OTRS ID and create a ticket. You do not have an OTRS ID yet? Register [here for free](#).



# Chapter 1. Feature List

## 1. Basic Mode

It's the default mode for this package.

In basic mode it is possible to define a master ticket and some slaves tickets that follows master tickets 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

### Warning

In basic mode Master Slave status can't be changed.

## 2. Advanced Mode

The advanced mode allows to make the use of the Master Slave relations more flexible by adding support to the following actions:

- New Master  
Slave tickets can be changed into new masters.
- Slave of Ticket# \*\*\*\*\*  
Master tickets can be turned into slaves tickets of other master ticket.
- Unset Master  
Master tickets can become normal tickets.
- Unset Slave  
Slave tickets can become normal tickets.

### Note

Above features can be enabled or disabled on SysConfig, for more information take a look in the "Configuration" chapter.

### 2.1. MasterSlave Screen

A new button "MasterSlave" is set to the ticket menu in the ticket zoom screen.

---

## Note

While this screen is available also in Basic Mode, the "Master Ticket" field is not available, it's only in advanced mode where this screen will let the change of the Master Slave status.

### 2.2. Bulk Action

If Advanced mode is enabled, bulk action screen enables the "Master Ticket" field to allow the change of the MasterSlave status.

### 3. Replace customer's from Master ticket

When a new article is created on the Master ticket, the Slave tickets replicates the article using the customer from the Master and not the own from the Slave.

Enabled article types defined in the SysConfig option 'ReplaceCustomerRealName-OnSlaveArticleTypes' will retain the customer that is set in the Slave tickets for Master ticket replicated articles.



## Chapter 2. System Requirements

### 1. Framework

The following OTRS framework is required:

- 6.0.19

### 2. Packages

The following packages are required:

- None

### 3. Operating System

This package requires one of the following operating systems:

- None

### 4. Third Party Software

This third party software is required to use this package:

- None.

## Chapter 3. Installation

The following instructions explain how to install the package.

### 1. Admin Interface

Please use the following URL to install the package utilizing the Admin Interface (please note that you need to be in the "admin" group).

<http://localhost/otrs/index.pl?Action=AdminPackageManager>

### 2. Command Line

Whenever you cannot use the Admin Interface for whatever reason, you may use the following command line tool ("bin/otrs.Console.pl Admin::Package::Install") instead.

```
shell> bin/otrs.Console.pl Admin::Package::Install /path/to/OTRSMasterSlave-6.0.15.opm
```



## Chapter 4. Configuration

The package can be configured via the SysConfig in the Admin Interface. The following configuration options are available:

### 1. **ACLKeysLevel3::Actions###888-OTRSMasterSlave.**

Navigation: Core::Ticket::ACL.

Defines which items are available for 'Action' in third level of the ACL structure.

### 2. **DashboardBackend###0900-TicketMaster.**

Navigation: Frontend::Agent::View::Dashboard.

Parameters for the dashboard backend of the master tickets overview of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.

### 3. **DashboardBackend###0910-TicketSlave.**

Navigation: Frontend::Agent::View::Dashboard.

Parameters for the dashboard backend of the slave tickets overview of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.

### 4. **DynamicFields::Driver###MasterSlave.**

Navigation: Core::DynamicFields::DriverRegistration.

DynamicField backend registration.

### 5. **Frontend::Module###AdminDynamicFieldMasterSlave.**

Navigation: Frontend::Admin::ModuleRegistration.

Frontend module registration for the agent interface.

### 6. **Frontend::Module###AgentTicketMasterSlave.**

Navigation: Frontend::Agent::ModuleRegistration.

Frontend module registration for the agent interface.

### 7. **Frontend::Navigation###AdminDynamicFieldMasterSlave###004-OTRSMasterSlave.**

Navigation: Frontend::Admin::ModuleRegistration::MainMenu.

Main menu item registration.

### 8. **Frontend::Navigation###AgentTicketMasterSlave###004-OTRSMasterSlave.**

Navigation: Frontend::Agent::ModuleRegistration::MainMenu.

Main menu item registration.

## **9. Loader::Module::AdminDynamicFieldMasterSlave###004-OTRS-MasterSlave.**

Navigation: Frontend::Admin::ModuleRegistration::Loader.

Loader module registration for the agent interface.

## **10. Loader::Module::AgentTicketMasterSlave###004-OTRSMasterSlave.**

Navigation: Frontend::Agent::ModuleRegistration::Loader.

Loader module registration for the agent interface.

## **11. MasterSlave::AdvancedEnabled.**

Navigation: Core::MasterSlave.

Enables the advanced MasterSlave part of the feature.

## **12. MasterSlave::DynamicField.**

Navigation: Core::MasterSlave.

Defines dynamic field name for master ticket feature.

## **13. MasterSlave::FollowUpdatedMaster.**

Navigation: Core::MasterSlave.

Enables the feature that slave tickets follow the master ticket to a new master in the advanced MasterSlave mode.

## **14. MasterSlave::ForwardSlaves.**

Navigation: Core::MasterSlave.

Enables the feature to forward articles from type 'forward' of a master ticket to the customers of the slave tickets. By default (disabled) it will not forward articles from type 'forward' to the slave tickets.

## **15. MasterSlave::KeepParentChildAfterUnset.**

Navigation: Core::MasterSlave.

Enables the feature to keep parent-child link after unset of the MasterSlave state in the advanced MasterSlave mode.

## **16. MasterSlave::KeepParentChildAfterUpdate.**

Navigation: Core::MasterSlave.

Enables the feature to keep parent-child link after change of the MasterSlave state in the advanced MasterSlave mode.

## **17. MasterSlave::UnsetMasterSlave.**

Navigation: Core::MasterSlave.

Enables the feature to unset the MasterSlave state of a ticket in the advanced MasterSlave mode.

## **18. MasterSlave::UpdateMasterSlave.**

Navigation: Core::MasterSlave.

Enables the feature to change the MasterSlave state of a ticket in the advanced MasterSlave mode.

## **19. PreApplicationModule###AgentPreMasterSlave.**

Navigation: Core::MasterSlave.

This module activates Master/Slave field in new email and phone ticket screens.

## **20. ReplaceCustomerRealNameOnSlaveArticleTypes.**

Group: MasterSlave, Subgroup: Frontend::Agent::Ticket::ViewMasterSlave.

This setting is deprecated and will be removed in further versions of OTRSMasterSlave.

## **21. ReplaceCustomerRealNameOnSlaveArticleCommunicationChannels.**

Group: MasterSlave, Subgroup: Frontend::Agent::Ticket::ViewMasterSlave.

Specifies the different article communication channels where the real name from Master ticket will be replaced with the one in the Slave ticket.

## **22. Ticket::EventModulePost###MasterSlave.**

Navigation: Core::Event::MasterSlave.

Registration of the ticket event module.

## **23. Ticket::Frontend::AgentTicketMasterSlave###ArticleTypeDefault.**

Navigation: Frontend::Agent::View::TicketMasterSlave.

Defines the default type of the note in the ticket MasterSlave screen of a zoomed ticket in the agent interface.

## **24. Ticket::Frontend::AgentTicketMasterSlave###ArticleTypes.**

Navigation: Frontend::Agent::View::TicketMasterSlave.

Specifies the different note types that will be used in the system.

## **25. Ticket::Frontend::AgentTicketMasterSlave###Body.**

Navigation: Frontend::Agent::View::TicketMasterSlave.

Sets the default body text for notes added in the ticket MasterSlave screen of a zoomed ticket in the agent interface.

## **26. Ticket::Frontend::AgentTicketMasterSlave###HistoryComment.**

Navigation: Frontend::Agent::View::TicketMasterSlave.

Defines the history comment for the ticket MasterSlave screen action, which gets used for ticket history in the agent interface.

## **27. Ticket::Frontend::AgentTicketMasterSlave###HistoryType.**

Navigation: Frontend::Agent::View::TicketMasterSlave.

Defines the history type for the ticket MasterSlave screen action, which gets used for ticket history in the agent interface.

## **28. Ticket::Frontend::AgentTicketMasterSlave###InformAgent.**

Navigation: Frontend::Agent::View::TicketMasterSlave.

Shows a list of all the possible agents (all agents with note permissions on the queue/ticket) to determine who should be informed about this note, in the ticket MasterSlave screen of a zoomed ticket in the agent interface.

## **29. Ticket::Frontend::AgentTicketMasterSlave###InvolvedAgent.**

Navigation: Frontend::Agent::View::TicketMasterSlave.

Shows a list of all the involved agents on this ticket, in the ticket MasterSlave screen of a zoomed ticket in the agent interface.

## **30. Ticket::Frontend::AgentTicketMasterSlave###MasterSlave-Mandatory.**

Navigation: Frontend::Agent::View::TicketMasterSlave.

Sets if Master / Slave field must be selected by the agent.

## **31. Ticket::Frontend::AgentTicketMasterSlave###Note.**

Navigation: Frontend::Agent::View::TicketMasterSlave.

Allows adding notes in the ticket MasterSlave screen of a zoomed ticket in the agent interface.

## **32. Ticket::Frontend::AgentTicketMasterSlave###NoteMandatory.**

Navigation: Frontend::Agent::View::TicketMasterSlave.

Sets if note must be filled in by the agent. Can be overwritten by Ticket::Frontend::NeedAccountedTime.

## **33. Ticket::Frontend::AgentTicketMasterSlave###Owner.**

Navigation: Frontend::Agent::View::TicketMasterSlave.

Sets the ticket owner in the ticket MasterSlave screen of a zoomed ticket in the agent interface.

## **34. Ticket::Frontend::AgentTicketMasterSlave###OwnerMandatory.**

Navigation: Frontend::Agent::View::TicketMasterSlave.

Sets if ticket owner must be selected by the agent.

## **35. Ticket::Frontend::AgentTicketMasterSlave###Permission.**

Navigation: Frontend::Agent::View::TicketMasterSlave.

Required permissions to use the ticket MasterSlave screen of a zoomed ticket in the agent interface.

### **36. Ticket::Frontend::AgentTicketMasterSlave###Priority.**

Navigation: Frontend::Agent::View::TicketMasterSlave.

Shows the ticket priority options in the ticket MasterSlave screen of a zoomed ticket in the agent interface.

### **37. Ticket::Frontend::AgentTicketMasterSlave###PriorityDefault.**

Navigation: Frontend::Agent::View::TicketMasterSlave.

Defines the default ticket priority in the ticket MasterSlave screen of a zoomed ticket in the agent interface.

### **38. Ticket::Frontend::AgentTicketMasterSlave###RequiredLock.**

Navigation: Frontend::Agent::View::TicketMasterSlave.

Defines if a ticket lock is required in the ticket MasterSlave screen of a zoomed ticket in the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).

### **39. Ticket::Frontend::AgentTicketMasterSlave###Responsible.**

Navigation: Frontend::Agent::View::TicketMasterSlave.

Sets the responsible agent of the ticket in the ticket MasterSlave screen of a zoomed ticket in the agent interface.

### **40. Ticket::Frontend::AgentTicketMasterSlave###Responsible-Mandatory.**

Navigation: Frontend::Agent::View::TicketMasterSlave.

Sets if ticket responsible must be selected by the agent.

### **41. Ticket::Frontend::AgentTicketMasterSlave###RichTextHeight.**

Navigation: Frontend::Agent::View::TicketMasterSlave.

Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

### **42. Ticket::Frontend::AgentTicketMasterSlave###RichTextWidth.**

Navigation: Frontend::Agent::View::TicketMasterSlave.

Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

### **43. Ticket::Frontend::AgentTicketMasterSlave###Service.**

Navigation: Frontend::Agent::View::TicketMasterSlave.

Sets the service in the ticket MasterSlave screen of a zoomed ticket in the agent interface (Ticket::Service needs to be activated).

#### **44. Ticket::Frontend::AgentTicketMasterSlave###State.**

Navigation: Frontend::Agent::View::TicketMasterSlave.

If a note is added by an agent, sets the state of the ticket in the ticket MasterSlave screen of a zoomed ticket in the agent interface.

#### **45. Ticket::Frontend::AgentTicketMasterSlave###StateDefault.**

Navigation: Frontend::Agent::View::TicketMasterSlave.

Defines the default next state of a ticket after adding a note, in the ticket MasterSlave screen of a zoomed ticket in the agent interface.

#### **46. Ticket::Frontend::AgentTicketMasterSlave###StateType.**

Navigation: Frontend::Agent::View::TicketMasterSlave.

Defines the next state of a ticket after adding a note, in the ticket MasterSlave screen of a zoomed ticket in the agent interface.

#### **47. Ticket::Frontend::AgentTicketMasterSlave###Subject.**

Navigation: Frontend::Agent::View::TicketMasterSlave.

Sets the default subject for notes added in the ticket MasterSlave screen of a zoomed ticket in the agent interface.

#### **48. Ticket::Frontend::AgentTicketMasterSlave###TicketType.**

Navigation: Frontend::Agent::View::TicketMasterSlave.

Sets the ticket type in the ticket MasterSlave screen of a zoomed ticket in the agent interface (Ticket::Type needs to be activated).

#### **49. Ticket::Frontend::AgentTicketMasterSlave###Title.**

Navigation: Frontend::Agent::View::TicketMasterSlave.

Shows the title field in the ticket MasterSlave screen of a zoomed ticket in the agent interface.

#### **50. Ticket::Frontend::BulkModule###010-MasterSlave.**

Navigation: Frontend::Agent::View::TicketBulk::Module.

MasterSlave module for Ticket Bulk feature.

#### **51. Ticket::Frontend::MenuModule###480-MasterSlave.**

Navigation: Frontend::Agent::View::TicketZoom::MenuModule.

Shows a link in the menu to change the MasterSlave status of a ticket in the ticket zoom view of the agent interface.

## Chapter 5. Usage

### Note

Be sure to enable the Master Slave dynamic field in the ticket zoom screen, this help to see quickly the Master Slave status in the "Ticket Information" box.

Be aware 'Ticket::Acl::Module###1-Ticket::Acl::Module' is not compatible to use with this package.

### 1. Create a Master Ticket

At the end of this example a new master ticket will be created.

1. Open "New phone Ticket" screen
2. Fill all mandatory elements of the screen
3. From "Master Ticket" select "New Master Ticket" and create the ticket
4. Write down the new ticket number (\*\*\*\*\*)
5. Zoom into the ticket to check the Master Slave status

### 2. Create A Slave Ticket

At the end of this example a new slave ticket will be created.

1. Be sure you have at least one Master ticket on the system
2. Open "New phone Ticket" screen
3. Fill all mandatory elements of the screen
4. From "Master Ticket" select "Slave of Ticket# \*\*\*\*\*: <Title>" and create the ticket
5. Zoom into the ticket to check the Master Slave status

### 3. Make Slave ticket To Follow Master Updates

At the end of this example the slave ticket will have a different priority by changing the master.

### Note

For this example we will change the priority of the master ticket 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 or add a note.

For this example the two tickets created in the above examples will be used.

Make sure that both tickets have a different priority than the one to be set for this example we will set priority "1 very low", so the tickets should be in a priority greater than "1 very low".

1. Zoom into the master ticket
2. Open the priority screen by clicking on the "Priority" button in the ticket menu



3. Fill any mandatory field and set the priority to "1 very low" and submit
4. Zoom into the slave ticket (This can be done quickly from the master ticket linked items table).
5. Look at the priority of the slave ticket.

## 4. Change The Master Slave Status

By the end of this tests the Master Slave status of a ticket will be changed

### Note

Depending on the current Master Slave status of the tickets, the enabled settings and the status of other tickets, different options will be presented in the "Master Ticket" field.

Make sure to enable the following options in the "SysConfig" settings:

- MasterSlave::AdvancedEnabled
- MasterSlave::UnsetMasterSlave
- MasterSlave::UpdateMasterSlave
- MasterSlave::FollowUpdatedMaste

### Warning

The most important setting to be enable is "MasterSlave::AdvancedEnabled", if it is not enabled then the "Master Ticket" field will not appear

1. Zoom into the slave ticket
2. Open Master Slave screen by click in in the "Master Slave" button in the ticket menu
3. Fill any mandatory field
4. Expand "Master Field" dropdown, the following options will appear:
  - New Master Ticket
  - Unset Slave Ticket

### Note

If there are other master tickets, for example: ticket number \$\$\$\$\$, then other options will appear like "Slave of Ticket# \$\$\$\$\$: <Title>", one per master ticket.

5. Choose "Unset Slave Ticket" and submit
6. Take a look to the new Master status in the "Ticket Information" box

### Note

As an additional test a change in the priority on the original master can be made, now the changed of the original master will not affect the former slave ticket.



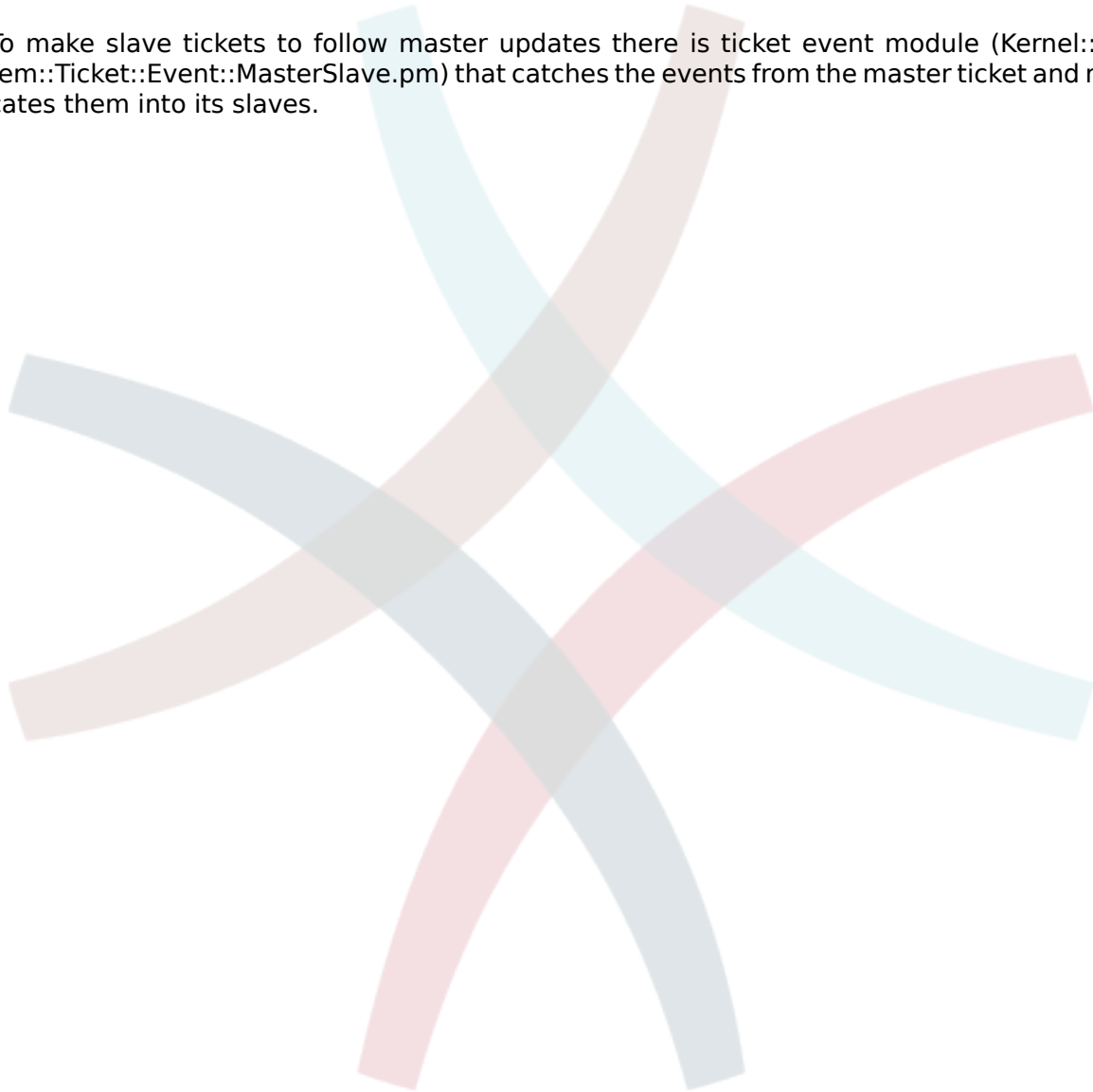
## Chapter 6. Technical Implementation Details

Master Slave relations are based in Parent Child link relations.

The package adds a new dynamic field type (MasterSlave). The creation of more fields of this type is not possible.

Master Slave information is stored in the `dynamic_field_value` table as any other dynamic field.

To make slave tickets to follow master updates there is ticket event module (`Kernel::System::Ticket::Event::MasterSlave.pm`) that catches the events from the master ticket and replicates them into its slaves.



## Chapter 7. File list

This list shows all included files and the referring permissions.

- (660) Kernel/Config/Files/XML/OTRSMasterSlave.xml
- (660) Kernel/Language/cs\_OTRSMasterSlave.pm
- (660) Kernel/Language/de\_OTRSMasterSlave.pm
- (660) Kernel/Language/es\_OTRSMasterSlave.pm
- (660) Kernel/Language/es\_MX\_OTRSMasterSlave.pm
- (660) Kernel/Language/hu\_OTRSMasterSlave.pm
- (660) Kernel/Language/id\_OTRSMasterSlave.pm
- (660) Kernel/Language/ja\_OTRSMasterSlave.pm
- (660) Kernel/Language/ms\_OTRSMasterSlave.pm
- (660) Kernel/Language/pl\_OTRSMasterSlave.pm
- (660) Kernel/Language/pt\_BR\_OTRSMasterSlave.pm
- (660) Kernel/Language/ru\_OTRSMasterSlave.pm
- (660) Kernel/Language/sr\_Cyrl\_OTRSMasterSlave.pm
- (660) Kernel/Language/sr\_Latn\_OTRSMasterSlave.pm
- (660) Kernel/Language/sw\_OTRSMasterSlave.pm
- (660) Kernel/Language/th\_TH\_OTRSMasterSlave.pm
- (660) Kernel/Language/zh\_CN\_OTRSMasterSlave.pm
- (660) Kernel/Modules/AdminDynamicFieldMasterSlave.pm
- (660) Kernel/Modules/AgentPreMasterSlave.pm
- (660) Kernel/Modules/AgentTicketMasterSlave.pm
- (660) Kernel/Output/HTML/TicketBulk/MasterSlave.pm
- (660) Kernel/Output/HTML/Templates/Standard/AdminDynamicFieldMasterSlave.tt
- (660) Kernel/Output/HTML/Templates/Standard/AgentTicketMasterSlave.tt
- (660) Kernel/System/DynamicField/Driver/MasterSlave.pm
- (660) Kernel/System/Ticket/Event/MasterSlave.pm
- (660) scripts/test/DynamicField/MasterSlave/EditFieldRender.t
- (660) scripts/test/DynamicField/MasterSlave/ValueSet.t
- (660) scripts/test/Selenium/Agent/AgentTicketCreateMasterSlave.t

- (660) scripts/test/Selenium/Agent/AgentTicketMasterSlave.t
- (660) scripts/test/Selenium/Output/TicketBulk/MasterSlave.t
- (660) scripts/test/Ticket/Event/MasterSlave/ReplaceCustomerRealName.t
- (660) scripts/test/Ticket/Event/MasterSlave.t
- (660) var/httpd/htdocs/js/Core.Agent.Admin.DynamicFieldMasterSlave.js
- (660) var/httpd/htdocs/js/Core.Agent.TicketMasterSlave.js
- (660) var/packagesetup/OTRSMasterSlave.pm



## Chapter 8. Tests

This module has been tested on the current state of the art in quality.

### 1. Test Cases

To tests this package please follow the examples described in the Usage section, all the tests cases should return the expected results defined at the beginning of each example.

### 2. Unit Tests

To ensure the quality of the module, several so-called unit tests were created, to test the functionalities of this module. These unit tests can be run via command line.

**ATTENTION:** Please never run unit tests on a productive system, since the added test data to the system will no longer be removed. Always use a test system.

Run the package specific unit tests

To run only the unit test which will be delivered with this package, use the following command on the command line:

```
shell> perl bin/otrs.Console.pl Dev::UnitTest::Run --test Ticket/Event/MasterSlave
shell> perl bin/otrs.Console.pl Dev::UnitTest::Run --directory DynamicField/MasterSlave
```

Selenium unit tests

To run the Selenium unit tests, make sure you have a complete Selenium environment setup in your Kernel/Config.pm file and a running a WebDriver.

```
shell> perl bin/otrs.Console.pl Dev::UnitTest::Run --test AgentTicketCreateMasterSlave --test
AgentTicketMasterSlave
shell> perl bin/otrs.Console.pl Dev::UnitTest::Run --test TicketBulk/MasterSlave
shell> perl bin/otrs.Console.pl Dev::UnitTest::Run --test Ticket/Event/MasterSlave/
ReplaceCustomerRealName
```

Run all available unit tests

To run all available unit tests, use the following command on the command line:

```
shell> perl bin/otrs.Console.pl Dev::UnitTest::Run
```

## Chapter 9. Change Log

### **6.0.15 / 2019-05-18 05:25:46**

- - Updated framework files to OTRS 6.0.19.

### **6.0.14 / 2019-04-06 05:32:18**

- - Updated framework files to OTRS 6.0.18. - Updated translations.

### **6.0.13 / 2019-02-23 06:04:19**

- - Updated framework files to OTRS 6.0.17.

### **6.0.12 / 2019-01-15 02:37:35**

- - Fixed bug#14350 ReplaceCustomerRealName is not working. - Updated framework files to OTRS 6.0.16. - Updated copyright.

### **6.0.11 / 2018-12-11 01:59:50**

- - Code cleanup. - Updated framework files to OTRS 6.0.15.

### **6.0.10 / 2018-11-02 22:02:11**

- - Updated translations, thanks to all translators. - Updated framework files to OTRS 6.0.13
- Fixed: Generic Agent log errors when master ticket is not available (bug#14082).

### **6.0.9 / 2018-09-15 07:24:45**

- - Updated translations, thanks to all translators. - Updated framework files to OTRS 6.0.11.
- Updated license information.

### **6.0.8 / 2018-07-24 10:11:09**

- - Added missing Change log. - Fixed: It is old setting values in the System Configuration of MasterSlave view (bug#13896).

### **6.0.7 / 2018-07-17 23:27:24**

- - Updated framework files to OTRS 6.0.9. - Fixed: Master / Slave dynamic field add screen is not modern (bug#13740).

### **6.0.6 / 2018-06-05 23:24:12**

- - Updated translations, thanks to all translators. - Updated framework files to OTRS 6.0.8.

### **6.0.5 / 2018-04-24 23:24:37**

- - Updated translations, thanks to all translators. - Updated framework files to OTRS 6.0.7.

### **6.0.4 / 2018-03-07 03:00:58**

- - Updated translations, thanks to all translators. - Updated framework files to OTRS 6.0.6.

---

**6.0.3 / 2018-02-06 20:10:24**

- - Updated translations, thanks to all translators. - Updated framework files to OTRS 6.0.5. - Fixed bug#12951 - Untranslated strings in TicketZoom screen. - Fixed bug#10415 - Rich-TextWidth and RichTextHeight settings missing. - Fixed bug#13257 - Legacy default subject in AgentTicketMasterSlave screen. - Fixed bug#8957 - MasterTicket field in MasterSlave action can't be set as mandatory. - Fixed bug#13306 - internal mail is sent to customer.

**6.0.2 / 2017-12-13 02:29:41**

- - Updated translations, thanks to all translators. - Updated framework files to OTRS 6.0.3. - Improved wording, thanks to Balázs Úr.

**6.0.1 / 2017-11-15 03:00:49**

- - Updated documentation (Config chapter). - Updated translations, thanks to all translators.

**5.0.92 / 2017-11-08 07:05:29**

- - Updated translations, thanks to all translators.

**5.0.91 / 2017-10-24 04:40:00**

- - Initial version for OTRS 6. - Updated translations, thanks to all translators. - Replaced UserFistname, UserLastname in favor of a more flexible UserFullname, thanks to Dian Tong Software.