KIX Pro Add-On Modules

KIX 17 Admin Manual - EN

Issued on: 13.02.2024

Table of contents

1 Anonymization 11

1.1 Installation of Anonymization Add-On Module ... 12

1.1.1 Prerequisite ... 12

1.1.2 Package Installation ... 12

1.2 Configuration of Anonymization Add-On Module ... 13

1.2.1 Anonymous Agent ... 13

1.2.2 Attributes for Anonymous Agents ... 13

1.2.3 Email Domain as Anonymous Agent ... 14

1.2.4 Creating Anonymous Agent if Necessary ... 14

1.2.5 Anonymous Customer ID ... 15

1.2.6 Match Criteria for Customer ID ... 15

1.2.7 Anonymous Customer ID by Backend ... 16

1.2.8 Match Criteria for Customer ID by Backend ... 16

1.2.9 Anonymous Customer User ... 16

1.2.10 Match Criteria for Customer User ... 17

1.2.11 Anonymous Customer User By Backend ... 17

1.2.12 Match Criteria for Customer User By Backend ... 17

1.2.13 Email as Anonymous Customer User ... 18

1.2.14 Anonymized History Comments ... 18

1.2.15 Article Email Addresses ... 19

1.2.16 Dynamic Field for Anonymization Flag ... 19

1.2.17 Ignoring Tickets with Anonymization Flag ... 20

1.3 Use of Anonymization ... 21

1.3.1 Configuring a Generic Agent ... 21

1.3.2 Anonymizing Tickets via Script ... 21

1.3.3 Example of an Anonymized Ticket ... 21

2 Field Service 24

2.1 Installation of Field Service Add-On Module ... 25

2.1.1 Prerequisites ... 25

2.1.2 Package Installation ... 25

2.1.3 New Group ... 25

2.2 Installation of App on a Mobile Device ... 26

2.2.1 Prerequisite ... 26

2.2.2 Installing the App ... 26

2.2.3 Configuring the Server URL ... 26

2.2.4 Registering Device ID in KIX ... 27
2.3 Configuration ... 28
2.3.1 Widget "Map view" in Kanban sidebar ... 28
2.3.2 Region Specification for Google Maps ... 29
2.3.3 Google API Key ... 29
2.3.4 Maximum Search Depth ... 29
2.3.5 Relevant CI Class ... 30
2.3.6 Relevant CI Class Attributes ... 30
2.3.7 Relevant Customer User Attributes ... 30
2.3.8 Cache Time To Live ... 30
2.3.9 Emphasizing Colors ... 31
2.3.10 Map Settings for Kanban View Configuration ... 31
2.3.11 Tour State Workflow ... 31
2.3.12 MobileDataSet: Maximum Idle Time ... 32
2.3.13 MobileDataSet: Output Directory ... 32
2.3.14 MobileDataSet: Included Attachments ... 32
2.3.15 MobileDataSet: Included Linked Objects ... 32
2.3.16 MobileDataSet: Device ID Attributes ... 32
2.3.17 MobileDataSet: Type of Entry in Ticket History for New, Synchronized Articles ... 33
2.3.18 MobileDataSet: Comment for Entry in Ticket History for New, Synchronized Articles ... 33
3 Inventory Add-On Module, Baramundi Backend 34
3.1 Baramundi Backend Installation ... 35
3.1.1 Prerequisites ... 35
3.1.2 Package Installation ... 35
3.2 Baramundi Backend Configuration ... 36
3.2.1 Source ... 36
3.2.2 "Baramundi" Backend ... 37
3.2.3 "BaramundiMD" Backend (Mobile Devices) ... 38
3.2.4 InventoryContentXPath for Baramundi Backend ... 40
3.2.5 Xpath Structure and Examples ... 41
3.2.6 Example ... 41
4 Inventory Add-On Module, DB Backend 47
4.1 DB Backend Installation ... 48
4.1.1 Prerequisites ... 48
4.1.2 Installation ... 48
4.2 DB Backend Configuration ... 49
4.2.1 Sources ... 49

4.2.2 Source – Backend – Mapping ... 49
4.2.3 Configuration in Config.pm ... 49
4.2.4 Inventory Content XPath ... 51
5 Inventory Add-On Module, OPSI Backend 52
5.1 OPSI Backend Installation ... 53
5.1.1 Prerequisites ... 53
5.1.2 Package Installation ... 53
5.2 OPSI Backend Configuration ... 54
5.2.1 Source ... 54
5.2.2 Source Backend Mapping ... 54
5.2.3 Source Parameters ... 54
5.2.4 SSL Verification ... 55
5.2.5 InventoryContentXPath Structure ... 55
5.3 Example of ConfigItem Class ... 56
6 Inventory Add-On Module, VMWare Backend 61
6.1 VMWare Backend Installation ... 62
6.1.1 Prerequisites ... 62
6.1.2 Package Installation ... 62
6.2 VMWare Backend Configuration ... 63
6.2.1 Basics ... 63
6.2.2 SysConfig Options ... 64
6.2.3 Source ... 64
6.2.4 Source Backend Mapping ... 64
6.2.5 Source Parameters ... 64
6.2.6 Alternative ObjectType Mapping ... 65
6.2.7 ObjectType Fallback ... 65
6.2.8 SSL Verification ... 65
6.2.9 InventoryContentXPath Structure ... 66
6.2.10 Example of ConfigItem Classes for VMWare Backend ... 81
7 ITSM Standardworkflows 82
7.1 Installation of ITSM-Standard Workflows Add-On Module ... 83
7.1.1 Prerequisites ... 83
7.1.2 Package Installation ... 83
7.2 Ticket Types ... 84
7.2.1 Relevant Ticket States ... 84
7.2.1.1 Workflow "Incident" + "Incident::Disaster" ... 86

7.2.1.2 Workflow "Service Request" + "Information Request" ... 87
7.2.1.3 Workflow "Problem" ... 88
7.2.1.4 Workflow "RfC::Standard" ... 89
7.2.2 Saving the Closure Code ... 89
7.2.3 Ticket Action "Record Workaround" ... 90
7.2.4 Ticket Action "Close" ... 90
7.3 Configuration of ITSM-Standard Workflows ... 91
8 Kanban Board 95
8.1 Installation of Kanban Board Add-On Module ... 96
8.1.1 Prerequisites ... 96
8.1.2 Package Installation ... 96
8.1.3 New Groups ... 96
8.1.4 New Tables ... 96
8.2 Configuration of Kanban Board Add-On Module ... 97
8.2.1 Widget "Tickets to dispatch" in Kanban Sidebar ... 98
8.2.2 Normalization of Column Widths ... 98
8.2.3 Ticket Attributes in Head Area of a Ticket Card ... 98
8.2.4 Ticket Attributes in Content Area of a Ticket Card ... 98
8.2.5 Dynamic Links for Ticket Attributes ... 99
8.2.6 Dummy Attributes ... 99
8.2.7 Progress Bar for Ticket Attributes ... 99
8.2.8 Meta States as Columns ... 99
8.2.9 Ticket Type Colors ... 100
8.2.10 Ticket-Type-Specific CSS Classes ... 100
8.2.11 Dynamic Fields as Swimlane Attributes ... 100
8.2.12 Number of Tickets in Kanban Board ... 101
8.2.13 Access to "Change" Popup ... 101
8.2.14 Setting the Required Lock ... 101
8.2.15 Showing the "Incident State" of Ticket Service ... 101
8.2.16 Changing Different Ticket Attributes and Dynamic Fields when Moving a Ticket in the Kanban Board ... 102
8.2.17 Next Possible State Types ... 102
8.2.18 Adding a Note when Moving a Ticket in the Kanban Board ... 103
8.2.19 Note Field as Mandatory Field ... 103
8.2.20 Default Values for Note Subject and Note Body ... 103
8.2.21 Size of the Note Input Field ... 104
8.2.22 Default Article Type ... 104
8.2.23 Possible Article Types ... 104
8.2.24 History Type ... 105
8.2.25 History Comment ... 105

9 KIX Connect 106
9.1 Installation of KIX Connect Add-On Module 107
9.1.1 Prerequisites 107
9.1.2 Installation 107
9.1.3 New Dynamic Fields 107
9.1.4 New ACLs 108
9.2 Generic Invoker 110
9.2.1 Use 110
9.3 Expanded Transport Options 114
9.3.1 Additional Headers 114
9.3.2 Proxy Settings 114
9.3.3 Configuration SSL 114
9.4 Dealing with Large Data Volumes 116
9.4.1 Caching Processed Datasets 116
9.4.2 Delaying/Omitting IncidentStateRecalc 117
9.4.3 Asynchronous Execution of Calls 118
9.5 Pre-Configured Web Services 119
9.5.1 KIX2KIX 120
9.5.1.1 Prerequisites 120
9.5.1.2 Activating "KIX2KIXRequester" Web Service on Requester System 120
9.5.1.3 Activating "KIX2KIXProvider" Web Service on Provider System 123
9.5.1.4 Creating a New Ticket from Requester KIX in Provider KIX 124
9.5.1.5 Creating a New Article in an External Ticket 125
9.5.1.6 Changing the Priority in an External Ticket 126
9.5.2 KIX2JIRA 127
9.5.2.1 Activating "KIX2JIRA" Web Service 127
9.5.2.2 Activating "KIX2JIRA_Attachment" Web Service 130
9.5.2.3 "JIRA information" Sidebar Widget 132
9.5.2.4 Creating a Ticket in KIX and Transmitting Data to JIRA 133
9.5.2.5 Changing a JIRA Process Title 134
9.5.2.6 Transmitting a Note in KIX as a JIRA Comment 134
9.5.3 X2i-doit 135
9.5.3.1 Activating "KIX2i-doit" Web Service 135
9.5.3.2 "Affected CIs" Sidebar Widget 137
9.5.3.3 Noting Affected CIs from i-doit on the Ticket 140
9.5.4 BaramundiSync 141
9.5.4.1 Activating "BaramundiSync" Web Service 141
9.5.5 VMWareAirWatchSync 145
9.5.5.1 Activating "VMWareAirWatchSync" Web Service 145

10 KPI Dashboard 149
10.1 Installation of KPI Dashboard Add-On Module 150
10.1.1 Prerequisites 150
10.1.2 Installation 150
10.2 Configuration of KPI Dashboard Add-On Module 152
10.2.1 KPI Area – Years Analyzed 153
10.2.2 Default Dashboard File 153
10.2.3 State Type Restriction 153
10.2.4 Archive Flag Restriction 153
10.2.5 Ignored Queues 154
10.2.6 Ignored Ticket Types 154
10.2.7 Ignored Services 154
10.2.8 Ignored SLAs 155
10.2.9 Escalation Base 155
10.2.10 Escalation Time Base 156
10.2.11 Data Stream – Directory for Data Preparation 156
10.2.12 Data Stream – Dynamic Fields 156
10.2.13 List of Available Ticket Attributes 156
10.2.14 List of Available KPI Attributes 157
10.2.15 List of Attributes Selected by Default 157
10.2.16 Database View 157
10.2.17 List of Available Database View Attributes 158
10.2.18 Symbols for the OutOfSLA Percentage 158
10.2.19 Settings for Elements in Online KPI Dashboard 159
10.2.20 Explanation of Default KPI Data 159
10.3 Administration of KPI Dashboard Add-On Module 162
10.3.1 Creating New KPI Dashboards 163
10.3.2 Configuring KPI Data Source in PowerPivot 165
10.3.3 Preparing Data Feeds 166
10.3.4 Preparing Data Feed using KIX GUI 166
10.3.5 Preparing Data Feed using KIX Console Script 166
10.3.6 Preparing Data Feed using Cron Jobs 167
10.4 Data Plugins for KPI Dashboard 168
10.5 Online KPI Dashboard 169
10.5.1 Opening Online KPI Dashboard 169
10.5.2 Settings for the Online KPI Dashboard 170
10.5.3 Filter for the Online KPI Dashboard 170

11 Maintenance Planner 171
11.1 Installation of Maintenance Planner Add-On Module 172
11.1.1 Prerequisites 172
11.1.2 Package Installation 172
11.1.3 New Access Control Lists (ACL) 172
11.1.4 Adapting the CI Classes 172
11.1.5 New Ticket Type 172
11.1.6 New Dynamic Fields 173
11.2 Configuration of Maintenance Planner Add-On Module 176
11.2.1 Classes for Time-Based Maintenance 176
11.2.2 Classes for Threshold-Based Maintenance 177
11.2.3 Relevant Deployment States 177
11.2.4 Checking Threshold-Based Maintenance 177
11.2.5 Link Types 178
11.2.6 Replacing the Current ConfigItem Version 178
11.2.7 Default Ticket Data 178
11.2.8 Ticket Attributes – ConfigItem-Attribute Mapping 178
11.2.9 Default Article Data 179
11.2.10 Scheduler Registration 179
11.2.11 Event for Validity Check 179
11.2.12 Validity Check – Parameters for Maintenance Planner 180
11.2.13 Validity Check – Parameters for Maintained ConfigItems 180
11.2.14 Event for Checking Value Update for Threshold-Based Maintenance 181
11.2.15 Writing Back Ticket Data to a ConfigItem 181
11.2.16 ConfigItem Attributes – Mapping of Dynamic Fields 181
11.2.17 Parameters for Searching for Similar Tickets 181
11.2.18 Permission for "Change" Popup 181
11.2.19 Setting Required Lock 181
11.2.20 Creating Maintenance Tickets 181
11.3 Use of Maintenance Planner 182
11.3.1 Adapting CI Classes for the Maintenance Planner 182
11.3.1.1 Adapting for Time-Based Maintenance 182
11.3.1.2 Adapting for Threshold-Based Maintenance 185
11.3.2 Composition of the Maintenance Intervals 189
12 MRO Bundle Add-On Module 190
12.1 Prelude 191
12.2 Installation 192

12.2.1 Installation under Debian / Ubuntu ... 192
12.2.2 Installation under OpenSUSE / SLES ... 192
12.2.3 Installation under CentOS / Red Hat ... 192
12.2.4 Notes on Upgrade ... 193
12.3 Components of KIX MRO Bundle ... 195
12.3.1 Kanban Board ... 195
12.3.2 Maintenance Planner ... 195
12.3.3 Field Service ... 195
12.3.4 KIX MRO Add-Ons ... 196
12.3.4.1 Generating CI Name ... 196
12.3.4.2 Event handler registration ... 196
12.3.4.3 Event handler configuration ... 196
12.3.4.4 Generating CI Number from Unique Attribute ... 197
12.3.4.5 Event handler registration ... 197
12.3.4.6 Event handler configuration ... 197
13 Security Management 198
13.1 General Information on Security Management ... 199
13.2 Installation of Security Management Add-On Module ... 200
13.2.1 Prerequisites ... 200
13.2.2 Package Installation ... 200
13.2.3 Migration of Version 17.4.x ... 201
13.2.3.1 Changes to Relevant SysConfig Settings ... 201
13.2.3.2 Modifications in Admin Area ... 201
13.3 Administration of Security Management ... 203
13.3.1 Calling up Tasks in KIX ... 203
13.3.2 New Objects ... 204
13.3.2.1 New Access Control List (ACL) ... 204
13.3.2.2 New Ticket Type ... 204
13.3.2.3 New Dynamic Fields ... 204
13.3.2.4 New CI Classes ... 205
13.3.2.5 New Ticket States ... 205
13.3.3 Configuration of Security Management ... 206
13.3.3.1 SysConfig Options ... 207
13.3.3.2 SOAP Interface – URL ... 207
13.3.3.3 SOAP Interface – User Name ... 207
13.3.3.4 SOAP Interface – Password ... 207
13.3.3.5 SOAP Interface – SSL ... 208
13.3.3.6 REST Interface – URL ... 208
13.3.3.7 REST Interface – User Name ... 208

13.3.3.8 REST Interface – Password 208
13.3.3.9 REST Interface – SSL 208
13.3.3.10 Source Name 209
13.3.3.11 verinice Backends 209
13.3.3.12 Synchronization – Relevant CI Classes 209
13.3.3.13 Synchronization – Attribute Mapping 209
13.3.3.14 Synchronization – Attribute Value Mapping 210
13.3.3.15 Synchronization – Link Type Mapping 210
13.3.3.16 Event Module for Ticket Changes 210
13.3.3.17 Ticket Type 210
13.3.3.18 Ticket State Workflow 211
13.3.3.19 Task Ticket – BSI Backend – Core Data 211
13.3.3.20 Task Ticket – BSI Backend – Article 211
13.3.3.21 Task Ticket – BSI Backend – Dynamic Fields – Mapping 211
13.3.3.22 Task Ticket – BSI Backend – Dynamic Fields – For Ticket Change 212
13.3.3.23 Task Ticket – BSIModern Backend – Core Data 212
13.3.3.24 Task Ticket – BSIModern Backend – Article 212
13.3.3.25 Task Ticket – BSIModern Backend – Dynamic Fields – Mapping 213
13.3.3.26 Task Ticket – BSIModern Backend – Dynamic Fields – For Ticket Change 213
13.3.3.27 Mapping KIX Identifier / verinice Identifier 213
13.3.3.28 Task Ticket – ISM Backend – Core Data 213
13.3.3.29 Task Ticket – ISM Backend – Article 214
13.3.3.30 Task Ticket – ISM Backend – Dynamic Fields – Mapping 214
13.3.3.31 Task Ticket – ISM Backend – Dynamic Fields – For Ticket Change 214
13.3.3.32 Task Ticket – Creation by Scheduler Module 214
13.3.3.33 Sidebar – Dynamic Fields 215
13.3.3.34 Ticket Tab – Dynamic Fields 215
13.3.3.35 Ticket Tab – Further Settings 215
13.3.4 Preparation for Use 216
13.3.4.1 Preparation for verinice 216
13.3.4.2 Synchronization 216
13.3.4.3 verinice Task Tickets 217
14 Telephone Integration 219
14.1 Configuration of Telephone Integration Add-On Module 220
14.1.1 Selection of Backend 220
14.1.2 Customer User PhoneID Type Mapping 220
14.1.3 Customer User PhoneID Type Mapping - Standard 221
14.1.4 Customer User PhoneID Mapping 221
14.1.5 Agent PhoneID Type 221
14.1.6 Incoming Calls – Filter 221
14.1.7 Recognition of Extension Number 222

14.1.8 Completion of Number Extension ... 222
14.1.9 Default Country Code ... 222
14.1.10 Settings for Outgoing Calls ... 222
14.1.11 Displaying Telephone Number Selection ... 223
14.1.12 Search Type for Searching for Agents and Contact Persons for Incoming Calls ... 223
14.1.13 Searching for Customers ... 223
14.2 Installation of Telephone Integration Add-On Module ... 224
14.2.1 Requirements for the Telephone System ... 224
14.2.1.1 Basic Requirements ... 224
14.2.1.2 Incoming calls ... 224
14.2.1.3 Outgoing calls ... 225
14.2.2 Package Installation ... 225
14.2.3 Integration in Your Telephone System ... 225
14.2.4 HTTP Backend – Configuration of ActionURLs ... 225
15 Time Accounting 227
15.1 Installation of Time Accounting Add-On Module ... 228
15.1.1 Prerequisites ... 228
15.1.2 Package Installation ... 228
15.2 Configuration of Time Accounting Add-On Module ... 229
15.2.1 CIC Dashlet and Ticket Sidebar Free Budgets ... 238
16 Liability Disclaimer for Add-On Modules 240
16.1 Liability for Contents ... 240
16.2 Liability for Links ... 240
16.3 Copyright ... 240
17 Purpose for which the use of KIX add-on modules is intended within a medical context 242

1 Anonymization

1.1 Installation of Anonymization Add-On Module

1.1.1 Prerequisite

- KIX Professional 17 – current version

1.1.2 Package Installation

Install the package "Ticket Anonymization" using "Paketverwaltung (Package Administration)" (Menu "Admin" area "System Administration") or using the command line.

1.2 Configuration of Anonymization Add-On Module

To configure settings for this function, select *"KIXTicketAnonymization"* group in SysConfig. After page has reloaded, select subgroup *"Core"*.

- Anonymous Agent (see page 13)
- Attributes for Anonymous Agents (see page 13)
- Email Domain as Anonymous Agent (see page 14)
- Creating Anonymous Agent if Necessary (see page 14)
- Anonymous Customer ID (see page 15)
- Match Criteria for Customer ID (see page 15)
- Anonymous Customer ID by Backend (see page 16)
- Match Criteria for Customer ID by Backend (see page 16)
- Anonymous Customer User (see page 16)
- Match Criteria for Customer User (see page 17)
- Anonymous Customer User By Backend (see page 17)
- Match Criteria for Customer User By Backend (see page 17)
- Email as Anonymous Customer User (see page 18)
- Anonymized History Comments (see page 18)
- Article Email Addresses (see page 19)
- Dynamic Field for Anonymization Flag (see page 19)
- Ignoring Tickets with Anonymization Flag (see page 20)

1.2.1 Anonymous Agent

- SysConfig setting: *KIXTicketAnonymization###AnonymousAgent*

Using this setting, you can define an anonymous agent login which replaces the owner and responsible person for anonymized tickets.

> **Please note**
>
> This setting is also used as a fallback. This means the agent must exist otherwise the agent data is not anonymized. Default value is: *"root@localhost"*. This setting is required.

1.2.2 Attributes for Anonymous Agents

- SysConfig setting: *KIXTicketAnonymization###AnonymousAgentAttribute*

Using this setting, you can specify an attribute for the current agent of a ticket, the value of which replaces the owner and responsible person in anonymized tickets.

Please note

This setting is deactivated by default.

1.2.3 Email Domain as Anonymous Agent

- SysConfig setting: *KIXTicketAnonymization###AnonymousAgentFromEMailDomain*

Using this setting, you can define whether the domain of the email attribute of an agent should be used as an anonymous agent. This setting is used when it is set to *"Yes"* and when the setting: "*KIXTicketAnonymization###AnonymousAgentFromEMailDomain"* has not been configured or when there is no attribute value for the current agent.

Please note

The setting is deactivated and set to *"No"* by default.

1.2.4 Creating Anonymous Agent if Necessary

- SysConfig setting: *KIXTicketAnonymization###AnonymousAgentCreate*

If this setting is active and the agent in the attribute of the setting: "*KIXTicketAnonymization###AnonymousAgentAttribute*" or the agent from the setting: "*KIXTicketAnonymization###AnonymousAgentFromEMailDomain*" does not exist, a new, "invalid" agent with the following attributes and values is created:

Attribute	Value
UserFirstName	Ano
UserLastName	Nymous

Attribute	Value
UserLogin	Attribute value of : "KIXTicketAnonymization###AnonymousAgentAttribut" or domain of setting: "KIXTicketAnonymization###AnonymousAgentFromEM" Please note: This setting is activated by default.
UserEmail	Attribute value or domain *"@noreply.local"*

Please note

Setting is deactivated by default.

1.2.5 Anonymous Customer ID

- SysConfig setting: KIXTicketAnonymization###AnonymousCustomerID

Using this setting, you can define which customer ID replaces the customer ID for anonymized tickets. If, instead of this, you would like to use a value from a customer attribute, then please use the setting: "KIXTicketAnonymization###AnonymousCustomerIDByCustomerBackend".

Please note

The setting is used as a fallback if setting "KIXTicketAnonymization###AnonymousCustomerIDByCustomerBackend" is not configured. If nothing is configured in either setting, the customer ID is not anonymized. Default value is "AnonymousCompany" and this setting is required.

1.2.6 Match Criteria for Customer ID

- SysConfig setting: KIXTicketAnonymization###AnonymizationCustomerID_MatchCriteria

Using this setting, you can define a regular expression. The customer ID is replaced with the value from setting "KIXTicketAnonymization###AnonymousCustomerID" only in tickets in which the current ticket customer ID matches the specified regular expression.

Please note

The setting is also the fallback for setting "KIXTicketAnonymization---AnonymousCustomerUserIDByCustomerBackend_MatchCriteria". Default value (.*) fits for everything. If the customer ID does not match, the contact person is also not anonymized.

1.2.7 Anonymous Customer ID by Backend

- SysConfig setting: KIXTicketAnonymization###AnonymousCustomerIDByCustomerBackend

Using this setting, you can specify a specific customer ID which replaces the customer ID in anonymized tickets, depending on the backend of the current ticket customer. The content can be a customer ID or ATTRIBUTE_AnyAttribute in order to use the value of the applicable attribute of the current customer user.

Please note

Setting is deactivated by default.

1.2.8 Match Criteria for Customer ID by Backend

- SysConfig setting: KIXTicketAn onymization###AnonymousCustomerIDByCustomerBackend_MatchCriteria

Using this setting, you can define a regular expression. The current customer ID is only replaced with the value from setting: "KIXTicketAnonymization###AnonymousCustomerIDByCustomerBackend" when it corresponds to the regular expression defined here.

Please note

Setting is deactivated by default. Default value is "(.*)"

1.2.9 Anonymous Customer User

- SysConfig setting: KIXTicketAnonymization###AnonymousCustomerUser

Using this setting, you can define a customer user login which replaces the customer user of the current ticket in anonymized tickets. If you would like to use a value from a customer user attribute, then please use setting: "AnonymousCustomerUserByBackend".

Please note

This setting is used as a fallback. If this setting is not specified, then the customer user data is not anonymized. Default value is "AnonymousUser". This setting is required.

1.2.10 Match Criteria for Customer User

- SysConfig setting: KIXTicketAnonymization###AnonymousCustomerUser_MatchCriteria

Using this setting, you can define a regular expression. The current customer user is then only replaced with the value from setting: "KIXTicketAnonymization###AnonymousCustomerUser" when it corresponds to the regular expression defined here.

Please note

This setting is also used as a fallback for the setting: "KIXTicketAnonymization###AnonymousCustomerUserByBackend_MatchCriteria". Default value is "(.*)".

1.2.11 Anonymous Customer User By Backend

- SysConfig setting: KIXTicketAnonymization###AnonymousCustomerUserByBackend

Using this setting, you can specify a specific customer user which replaces the customer user in anonymized tickets, depending on the backend of the current ticket customer. The content can be a customer user ID or ATTRIBUTE_AnyAttribute in order to use the value of the applicable attribute of the current customer user.

Please note

This setting is deactivated by default.

1.2.12 Match Criteria for Customer User By Backend

- SysConfig setting: KIXTicketAnonymization###AnonymousCustomerUserByBackend_MatchCriteria

Using this setting, you can define a regular expression. The current customer ID is then only replaced with the value from setting: "KIXTicketAnonymization###AnonymousCustomerUserByBackend" when it corresponds to the regular expression defined here.

Please note

Setting is deactivated by default. Default value is "(.*)"

1.2.13 Email as Anonymous Customer User

- SysConfig setting: KIXTicketAnonymization###AnonymousCustomerUserFrom

Using this setting, you can define whether the domain of the email attribute of a customer user should be used as an anonymous customer user. This setting is used when it is set to "Yes" and when the setting: "KIXTicketAnonymization###AnonymousCustomerUserByBackend" has not been configured or when there is no attribute value for the current agent.

Please note

The setting is deactivated and set to "No" by default.

1.2.14 Anonymized History Comments

- SysConfig setting: KIXTicketAnonymization###HistoryTypeAnonymousComments

Using this setting, you can define which comments should be contained in the history of anonymized tickets, and which text the comments contain. The following default values are provided:

Attribute	Value
CustomerUpdate	CustomerUpdate: comment replaced by CustomerUpdate KIXTicketAnonymization.
OwnerUpdate	OwnerUpdate: comment replaced by OwnerUpdate KIXTicketAnonymization.
ResponsibleUpdate	ResponsibleUpdate: comment replaced by KIXTicketAnonymization.
SendAgentNotification	SendAgentNotification: comment replaced by KIXTicketAnonymization.
SendAnswer	SendAnswer: comment replaced by KIXTicketAnonymization.

SendAutoReply	SendAutoReply: comment replaced by KIXTicketAnonymization.
SendCustomerNotification	SendCustomerNotification: comment replaced by KIXTicketAnonymization.
TicketLinkAdd	Added involved personen: comment replaced by KIXTicketAnonymization.

Please note

Setting is activated by default and is required.

1.2.15 Article Email Addresses

- SysConfig setting: KIXTicketAnonymization###ReplaceEmailAddresses

Using this setting, you can define an anonymous email address which can be used to replace the email addresses in articles of anonymized tickets when the article email address is not a system address.

Please note

Default value is "kix@localhost". This setting is required.

1.2.16 Dynamic Field for Anonymization Flag

- SysConfig setting: KIXTicketAnonymization###AnonymizationFlag

Using this setting, you can configure a dynamic field which is used to set the value of the anonymization flag.

Please note

This setting is activated by default. During the installation of this add-on, the dynamic field "Anonymized" was created so that you can use the provided configuration. It is possible that you will have to update your configuration if you would like to have the dynamic field shown in certain displays.

1.2.17 Ignoring Tickets with Anonymization Flag

- SysConfig setting: KIXTicketAnonymization###IgnoreTicketsWithAnonymizationFlag

Using this setting, you can specify that tickets which have already been anonymized can be ignored in future anonymization processes.

> **Please note**
>
> This setting is activated by default.

1.3 Use of Anonymization

There are two different options for anonymizing tickets. You can create a Generic Agent for this in the Admin area or use the command line script.

1.3.1 Configuring a Generic Agent

The Generic Agent for this function must be created manually. In the "Admin" area, select "Generic Agent" in the "System Administration" widget. In the "Generic Agent" area, click "Add job" button. In the individual widgets, now specify the details for the tickets that are to be anonymized.

Enter the following module designation in the "Execute Custom Module" widget:

```
Kernel::System::GenericAgent::KIXTicketAnonymization
```

Finally, click button "Submit". The job now appears in the list of defined jobs.

Click "Run this task" in column "Run now!", to execute the created job immediately.

1.3.2 Anonymizing Tickets via Script

There is also the possibility the anonymize tickets using command lines. Here, execute:

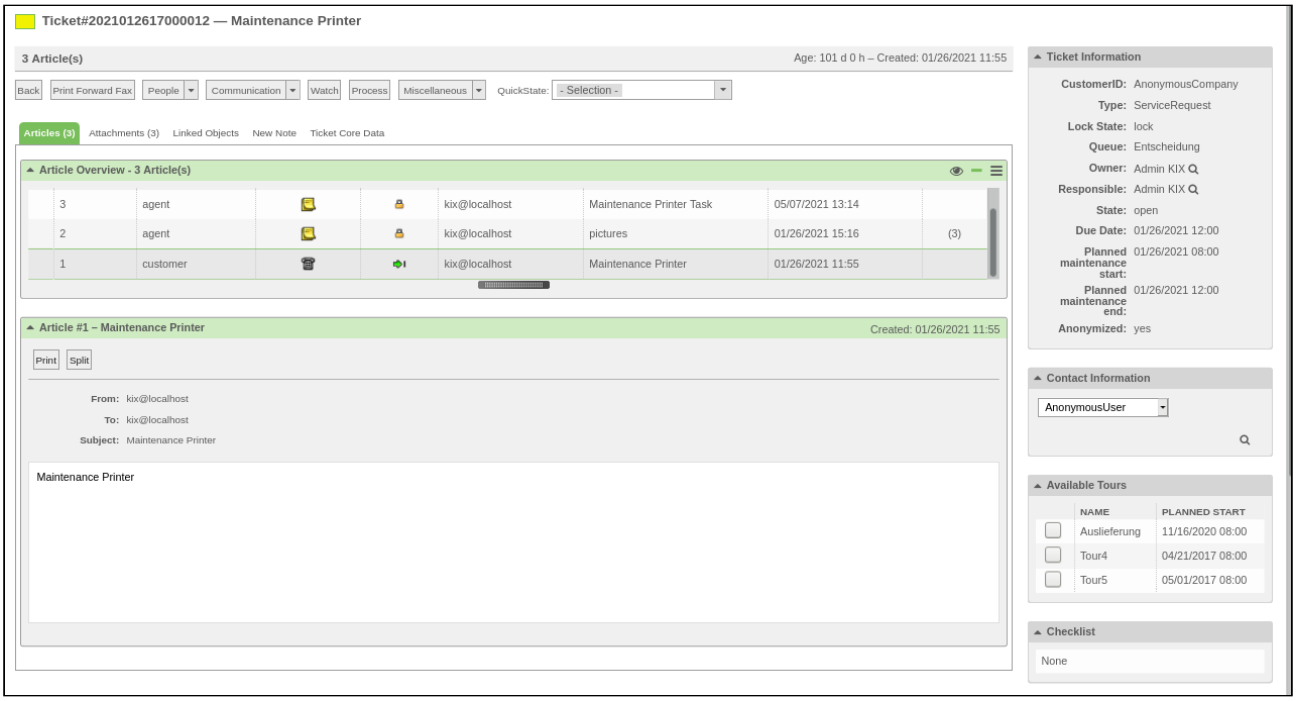
```
/bin/kix.Console.pl Admin::KIXTicketAnonymization::Anonymize
```

with one of the parameters "--user agent_login" or "--customer customer_user_login".

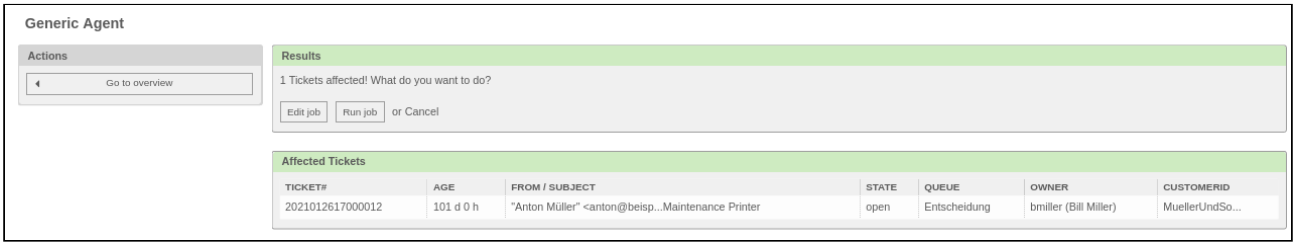
If an agent login has been specified, all tickets are anonymized that the agent is the owner of or responsible for, based on the information in the SysConfig. If a customer user login has been specified, all tickets of this customer user are anonymized based on the information in the SysConfig.

1.3.3 Example of an Anonymized Ticket

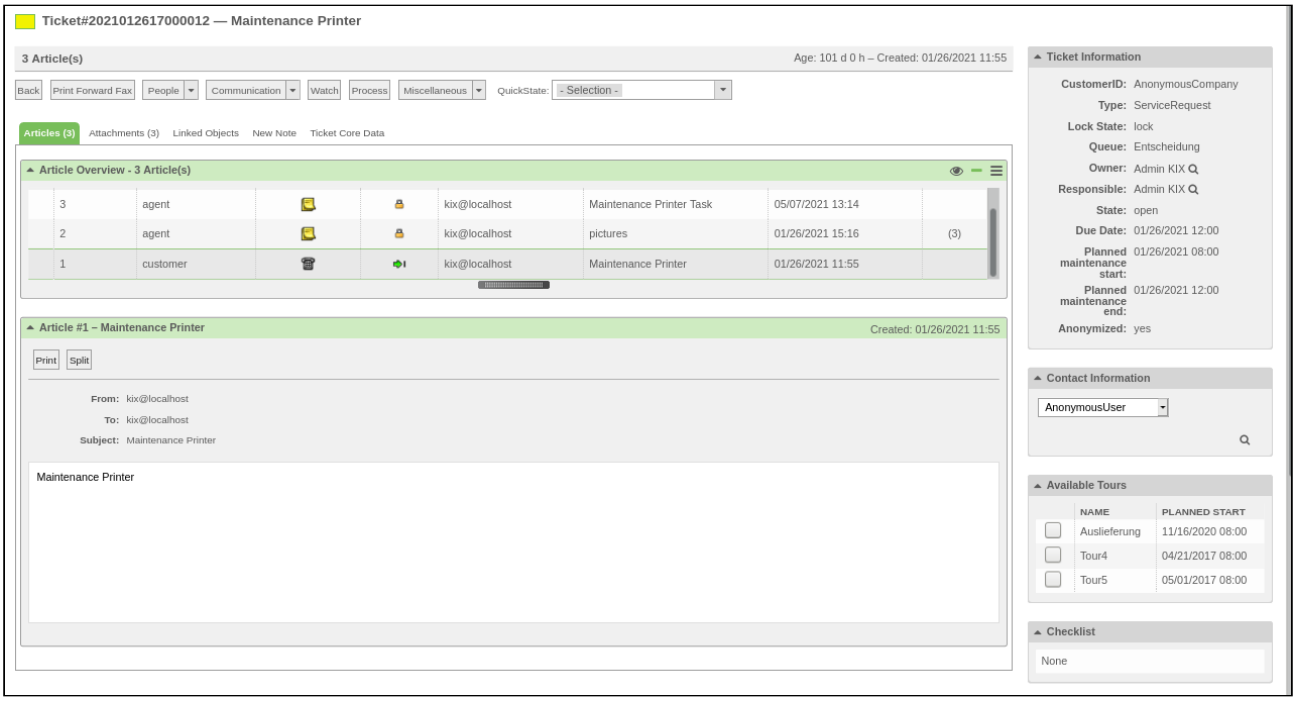
- Ticket before anonymization



- The job for anonymization is executed in Generic Agent.



- This is what the anonymized ticket looks like.


Ticket#2021012617000012 — Maintenance Printer
3 Article(s)
Age: 101 d 0 h – Created: 01/26/2021 11:55
Back
Print Forward Fax
People
Communication
Watch
Process
Miscellaneous
QuickState:
- Selection -
Articles (3)
Attachments (3)
Linked Objects
New Note
Ticket Core Data
Article Overview - 3 Article(s)
3 agent kix@localhost Maintenance Printer Task 05/07/2021 13:14
2 agent kix@localhost pictures 01/26/2021 15:16 (3)
1 customer kix@localhost Maintenance Printer 01/26/2021 11:55
Article #1 – Maintenance Printer
Created: 01/26/2021 11:55
Print
Split
From: kix@localhost
To: kix@localhost
Subject: Maintenance Printer
Maintenance Printer
Ticket Information
CustomerID: AnonymousCompany
Type: ServiceRequest
Lock State: lock
Queue: Entscheidung
Owner: Admin KIX
Responsible: Admin KIX
State: open
Due Date: 01/26/2021 12:00
Planned maintenance start: 01/26/2021 08:00
Planned maintenance end: 01/26/2021 12:00
Anonymized: yes
Contact Information
AnonymousUser
Available Tours
NAME
PLANNED START
Auslieferung 11/16/2020 08:00
Tour4 04/21/2017 08:00
Tour5 05/01/2017 08:00
Checklist
None

2 Field Service

2.1 Installation of Field Service Add-On Module

2.1.1 Prerequisites

- KIX Professional 17 – current version
- If already available: Uninstall the Kanban board.
    - The Field Service module comes with the Kanban board by default. If the Kanban board module is already installed, the Field Service module cannot be installed and the system issues an error message.

2.1.2 Package Installation

Install the package "KIXFieldService" using "Paketverwaltung (Package Administration)" (Menu "Admin" area "System Administration") or using the command line.

2.1.3 New Group

During installation, a new group "FieldService" is created. The group provides access to the "Kartenansicht (Card view)" widget in the Kanban Board sidebar. The initial user (default: root@localhost) has writing access for this group following installation.

2.2 Installation of App on a Mobile Device

The Field Service app is a mobile application (app) for Android devices in order to provide data from KIX offline on the end device for service technicians. This is to ensure that they can work in areas without an Internet connection. The service technicians can view the necessary data with the app and document their work. Once all work has been carried out, this documentation can be synchronized with the server again.

2.2.1 Prerequisite

To be able to use the app, the following prerequisites have to be met:

- A mobile end device (smartphone, tablet, etc.) with Android 4.0 or higher.
- Network connection between the mobile end device and KIX server for the download/upload.

2.2.2 Installing the App

The app is not available in the Google Play store. If you would like to download the app, enter

the following URL in the browser of the mobile end device:

"http://<FQDN KIX-Server>FQDN KIX-Server/kix-web/FieldService.apk"

Once downloaded, open the file: FieldService.apk.

Please note

In the Android device's system settings, you may have to change the setting for installing apps from "unknown sources".

2.2.3 Configuring the Server URL

So that the app can communicate with the KIX server, the KIX server URL has to be specified in the app. If you are starting up the app for the first time, you will be asked to enter the server URL.

Example: *"http://<FQDN KIX-Server>/kix/index.pl"'*

2.2.4 Registering Device ID in KIX

Once the server URL has been configured, the app displays a screen with a link "DeviceID anzeigen (Display DeviceID)".

If you tap on this link, you will see the device ID.

This device ID must be entered in KIX for the ConfigItem that the mobile end device represents as the "Seriennummer (Serial number)" attribute. This information is important so that the mobile data sets for the relevant mobile end device can be generated and transmitted.

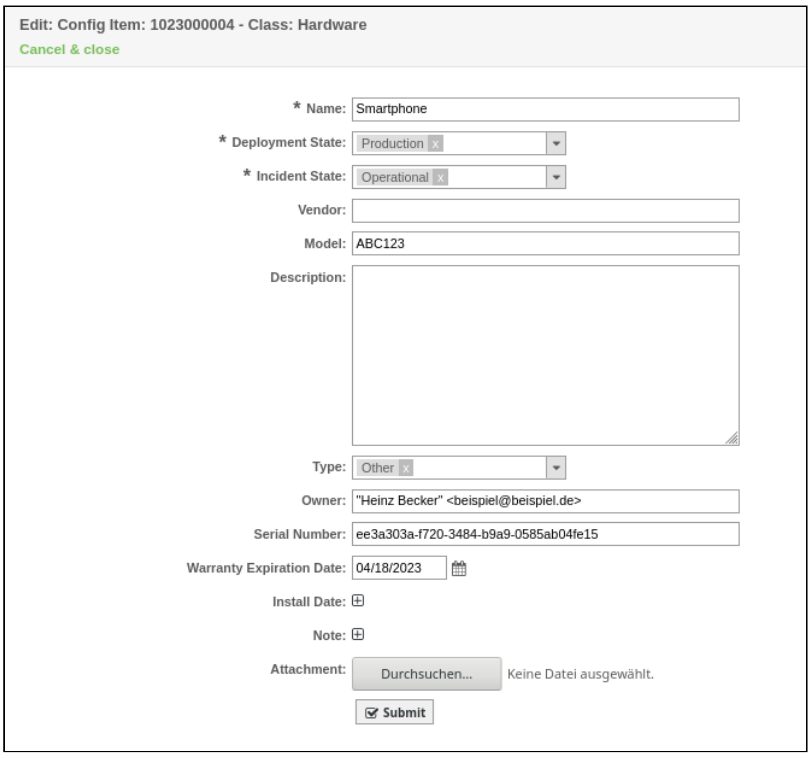


Figure: ConfigItem – Serial Number Attribute

2.3 Configuration

Once you have installed the package, if necessary, you can set the default values in the Admin area/ SysConfig. Here, select SysConfig group "KIXFieldService". After page has reloaded, select subgroups "Frontend::Agent::FieldService".

If you would like to configure settings for the module "Kanban-Board", select SysConfig groups "Kanban4KIX" and one of the subgroups "Frontend::Agent::KanbanView" or "Frontend::Agent:: AgentKanbanActionCommonPopup".

Default configuration is already contained in the package. You can adapt this in accordance with your wishes. The SysConfig settings are explained in the following sections.

- Widget "Map view" in Kanban sidebar (see page 28)
- Region Specification for Google Maps (see page 29)
- Google API Key (see page 29)
- Maximum Search Depth (see page 29)
- Relevant CI Class (see page 30)
- Relevant CI Class Attributes (see page 30)
- Relevant Customer User Attributes (see page 30)
- Cache Time To Live (see page 30)
- Emphasizing Colors (see page 31)
- Map Settings for Kanban View Configuration (see page 31)
- Tour State Workflow (see page 31)
- MobileDataSet: Maximum Idle Time (see page 32)
- MobileDataSet: Output Directory (see page 32)
- MobileDataSet: Included Attachments (see page 32)
- MobileDataSet: Included Linked Objects (see page 32)
- MobileDataSet: Device ID Attributes (see page 32)
- MobileDataSet: Type of Entry in Ticket History for New, Synchronized Articles (see page 33)
- MobileDataSet: Comment for Entry in Ticket History for New, Synchronized Articles (see page 33)

2.3.1 Widget "Map view" in Kanban sidebar

- SysConfig setting: Frontend::KanbanSidebarBackend###FieldServiceMap

This option activates the "Kartenansicht (Card view)" widget in the Kanban Board sidebar. The "Group" key contains the agent groups, separated by semicolons, which have access to the widget.

Please note

Default value for this group is "FieldService".

2.3.2 Region Specification for Google Maps

- SysConfig setting: FieldServiceMap::Region

Determines the region for Google Maps. This influences the behavior of functions such as determining geolocation data. The majority of regions are identical to the applicable country-specific top-level domains (see here ISO 3166-1), but there are some exceptions (e.g., "GB" instead of "UK" for Great Britain).

Please note

This setting is required. Default value is "DE".

2.3.3 Google API Key

- SysConfig setting: FieldServiceMap::GoogleApiKey

Here you can enter your Google API key. The key is required to use the FieldService card view. You can find more information here:

```
https://developers.google.com/maps/documentation/javascript/get-api-key
```

2.3.4 Maximum Search Depth

- SysConfig setting: FieldServiceMap::MaxLinkDepth

Using this setting, you define the search depth for linked CIs in order to obtain address data.

Please note

This setting is required and the default value is "3".

2.3.5 Relevant CI Class

- SysConfig setting: FieldServiceMap::CIClassWithAddress

This setting specifies the relevant CI class with the address data.

> **Please note**
>
> Default value is "Location".

2.3.6 Relevant CI Class Attributes

- SysConfig setting: FieldServiceMap:CIClassWithAddressAttributes

This setting determines the relevant CI class attributes which contain address data.

> **Please note**
>
> Default value is "Address".

2.3.7 Relevant Customer User Attributes

- SysConfig setting: FieldServiceMap::UserAddressFields

Using this setting, you define the relevant customer user attributes which contain the address data. This data is also used as a fallback if a ticket does not have any linked CIs or if no address data can be determined with linked CIs.

> **Please note**
>
> Default values are "UserStreet" , "UserZip" , "UserCity", and "UserCountry".

2.3.8 Cache Time To Live

- SysConfig setting: FieldServiceMap::GeoLocationCacheTTL

This setting determines the number of days that the geolocation data of an address are retained in the cache.

Please note

Default value is "30".

2.3.9 Emphasizing Colors

- SysConfig setting: FieldServiceMap::EmphasizeColor

Using this setting, you can determine the fill and border color of the map markings and Kanban ticket.

Please note

The default value is blue "#00B1D5" for the filling and "#005D70" for the border.

2.3.10 Map Settings for Kanban View Configuration

- SysConfig setting: Frontend::Output::FilterElementPre###OutputFilterAgentFieldServiceMapSettings

This setting determines the output filter provided by the map setting in the Kanban View configuration. The value for "DefaultMapCenter" can be an address or a longitude/latitude coordinate. The value for "DefaultZoom" should be a number between 1 and 18.

Please note

Default values are "51.163375 , 10.447683" (center of Germany) and "6" for the zoom.

2.3.11 Tour State Workflow

- SysConfig setting: FieldService::Tour###StateWorkflow

This setting defines the state workflow of a field service tour.

Please note

Only change when you are confident in what you are doing.

2.3.12 MobileDataSet: Maximum Idle Time

- SysConfig setting: FieldService::MobileDataSet###Directory

This setting determines the maximum idle time in the FieldService app. If the user does not work in the app within this time, the app logs the user off. The default value is 1800 seconds (30 minutes).

2.3.13 MobileDataSet: Output Directory

- SysConfig setting: FieldService::MobileDataSet###Directory

This setting specifies the output directory in which the MDS data is generated.

Please note

Only change when you are confident in what you are doing.

2.3.14 MobileDataSet: Included Attachments

- SysConfig setting: FieldService::MobileDataSet###IncludeAttachments

This settings defines the pre-setting for attachments of the objects included in a tour. When creating a tour, you can overwrite this setting in the dialog.

2.3.15 MobileDataSet: Included Linked Objects

- SysConfig setting: FieldService::MobileDataSet###IncludeLinkedObjects

This settings defines the pre-setting for directly linked objects included in a tour. When creating a tour, you can overwrite this setting in the dialog.

2.3.16 MobileDataSet: Device ID Attributes

- SysConfig setting: FieldService::MobileDataSets###CIAttributeDeviceID

This setting determines the CI attribute which contains the ID of the mobile end device. Default value is "SerialNumber".

2.3.17 MobileDataSet: Type of Entry in Ticket History for New, Synchronized Articles

- SysConfig setting: FieldService::MobileDataSet###HistoryType

This setting specifies the type of applicable history entry when a new article from the mobile end device is synchronized with the server. Default value is "AddNote".

2.3.18 MobileDataSet: Comment for Entry in Ticket History for New, Synchronized Articles

- SysConfig setting: FieldService::MobileDataSet###HistoryComment

This setting specifies the comment for a history entry when a new article from the mobile end device is synchronized with the server. Default value is "synchronized article from mobile device".

3 Inventory Add-On Module, Baramundi Backend

3.1 Baramundi Backend Installation

3.1.1 Prerequisites

- KIX Professional 17 – current version
- Baramundi 8.9 or higher

3.1.2 Package Installation

Install the package "kixpro-inventorysync-baramundi" using "Paketverwaltung (Package Administration)" (Menu "Admin" area "System Administration") or using the command line.

3.2 Baramundi Backend Configuration

The package contains an example configuration that you can adapt to your needs. To configure settings for this function, select "KIX Professional" group in SysConfig. After page has reloaded, select subgroup "ITSMConfigItem::InventorySync".

- Source (see page 36)
- "Baramundi" Backend (see page 37)
- "BaramundiMD" Backend (Mobile Devices) (see page 38)
- InventoryContentXPath for Baramundi Backend (see page 40)
- Xpath Structure and Examples (see page 41)
- Example (see page 41)

3.2.1 Source

- SysConfig setting: ITSMConfigItem::InventorySync###Sources

A source requires a named communication point. A hash is used for configuration. The hash key is the internal name of the source, for example "inventory1". The hash value is the display name for this source, for example "inventory server 1".
You can add several sources for each external inventory server.

Source backend mapping

- SysConfig setting: ITSMConfigItem::InventorySync###Backend

This setting defines the backend to be used. A hash is used for configuration. The hash key is the identifier of the source (in our example "inventory1"). For non-mobile devices, the hash value must be specified as "Baramundi" in the field "Content". For mobile devices, the value "BaramundiMD" must be entered in the field "Content".

Source parameters

- SysConfig setting: ITSMConfigItem::InventorySync###Parameters

In this setting, you have to enter the database connection string (DSN) in the following form:

DatabaseDSN=DBI:ODBC:<DataSource>;DatabaseUser=<uid>;DatabasePw=<pwd>;<additonal parameters>

Parameter	Benötigt	Beschreibung
DataSource	x	Name of Baramundi ODBC data source.
DatabaseUser	x	User name of the database user.
DatabasePw	x	Password of the database user.
Additional Parameters		Depends on configured data source. If the data source is an MSSQL server, you have to enter "Type=mssql" as an additional parameter.
UseInventoryType		Optional parameter to specify which type of inventory data should be used. If not specified, the data from the "Standard Template" is used. If you want to use WMI data, set the value to "*WMI*". **Hinweis** Please note: This parameter is not possible for mobile devices.

Note

Please note: A DSN-based ODBC connection string must be used. Other configurations will not work.

3.2.2 "Baramundi" Backend

By default, the following information is added to the data from the different hardware scans (standard and WMI):

- General Information
  - Manufacturer
  - ModelName
  - OperatingSystem
  - CurrentUser
  - PrimaryIP
  - PrimaryMAC
  - TimeZone

- MACList

3.2.3 "BaramundiMD" Backend (Mobile Devices)

The data of mobile end devices is managed differently in Baramundi to the data of non-mobile devices. This means the data structure provided by this backend differs significantly to that of the "Baramundi" backend. The following data is provided from Baramundi:

- Overview
  - General
  - Operating system
  - Network
- Device inventory
  - Device information
  - Restrictions
  - Security settings
  - Installed Wi-Fi configurations
- Installed apps (incl. system apps)

For a detailed list of the data delivered, please see the provided console script (see "InventoryContentXPath").

Standard hardware scans

The structure and content of hardware inventory data returned from the Baramundi backend is almost identical to the content of the tree view for each hardware inventory scan in Baramundi. It has the following form:

- Central Processor Unit(s)
  - <List of attributes for each CPU>
- Motherboard
  - <List of attributes for each motherboard>
- Memory
  - <List of attributes for each RAM slot>
- Video Adapter
  - <List of attributes for each video adapter>
- Monitor
  - <List of attributes for each monitor>
- Drives::Floppy
  - <List of attributes for each floppy drive>
- Drives::IDE Drives
  - <List of attributes for each IDE drive>

- Audio
  - <List of attributes for each audio device>
- Network
  - <List of attributes for each network device>
- Ports
  - <List of attributes for each port>

WMI Hardware Scans

In Baramundi, the structure of WMI scans differs from the standard scans. In InventorySync, the data is therefore mapped in a structure that is similar to the structure of the standard scans. It fundamentally has the following form, where the attributes of each item match the columns shown in Baramundi:

Central Processor Unit(s)

- Refers to the Baramundi node "Win32_Processor"
- <List of attributes for each CPU>

BIOS

- Refers to the Baramundi node "Win32_Bios"
- <List of attributes for each BIOS node>

Motherboard

- Refers to the Baramundi node "Win32_BaseBoard"
- <List of attributes for each motherboard>

Memory

- Refers to the Baramundi node "Win32_PhysicallyMemory"
- <List of attributes for each RAM slot>

Video Adapter

- Refers to the Baramundi node "Win32_VideoController"
- <List of parameters for each video adapter>

Monitor

- Refers to the Baramundi node "Win32_DesktopMonitor"
- <List of attributes for each monitor>

Drives::Floppy

- Refers to the Baramundi node "Win32_FloppyDrive"
- <List of attributes for each floppy drive>

Drives::CDROM

- Refers to the Baramundi node "Win32_CDROMDrive"

- <List of attributes for each CDROM drive>

Drives::PhysicalDisk

- Refers to the Baramundi node "Win32_DiskDrive"
- <List of attributes for each hard drive>

Drives::LogicalDisk

- Refers to the Baramundi node "Win32_LogicalDrive"
- <List of attributes for each logical drive>

Network

- Refers to the Baramundi node "Win32_NetworkAdapterConfiguration"
- <List of attributes for each network device

Ports

- Refers to the Baramundi node "Win32_PortConnector"
- <List of attributes for each port>

If Baramundi does not contain detailed attributes for a device, this device is ignored.

Software scans

Each software structure is identified by the unique software GUID (Globally unique identifier), and contains the following attributes:

- Name
- Producer
- Version

3.2.4 InventoryContentXPath for Baramundi Backend

The basic expansion of ConfigItem classes and the use of InventoryContentXPath parameters are described in Chapter "KIX SysConfig/KIX Professional/InventorySync". There is a console script to analyze the relevant data:

```
Admin::ITSM::InventorySync::DumpBaramundiData
```

This is particularly helpful to be able to determine the correct XPath. Using the parameter "--source", the ID of the source is specified with which the script is to communicate. A call-up with the parameter "--list-clients" displays a list of the devices in the configured source.

The parameter "--client-id", to which the name of the applicable device is transferred, generates the output of the determined data structure. From this, the relevant XPath information can then be created.

3.2.5 Xpath Structure and Examples

The Xpath for Baramundi has the following structure:

- Hardware Xpath: HW/<node>/<attribute>
- Software Xpath: SW/<guid>/<attribute>

Typical Xpaths for Baramundi may have the following appearance:

- HW/Central Processors(s)/CPU Brand Name to get the name attribute of each processor
- HW/Drives::IDE Drives/Drive Model to get the model attribute of each harddisk drive
- SW/.*?/Name to get the name of the software, regardless of its GUID

3.2.6 Example

The following example shows a ConfigItem class definition for the class "Computer" with some changes for the Baramundi backend based on standard hardware scans. Please note that these are not required changes. You can make your own changes. You must just ensure that the necessary ConfigItem attributes receive values during synchronization. Otherwise an error message will be displayed.

```
[
    {
        Key => 'ClientGroup',
        Name => 'verinice ClientGroup',
        Searchable => 1,
        Input => {
            Type => 'CIClassReference',
            ReferencedCIClassName => 'ClientGroup',
            ReferencedCIClassLinkType => 'Includes',
            ReferencedCIClassLinkDirection => 'Reverse',
            ReferencedCIClassReferenceAttributeKey => 'Name',
            Required => 0,
        },
    },
    {
        Key => 'Vendor',
        Name => 'Vendor',
        Searchable => 1,
        Input => {
            Type => 'Text',
```

```
            Size => 50,
            MaxLength => 50,
        },
    },
    {
        Key => 'Model',
        Name => 'Model',
        InventoryContentXPath => 'HW/Motherboard/Computer Brand Name',
        Searchable => 1,
        Input => {
            Type => 'Text',
            Size => 50,
            MaxLength => 50,
        },
    },
    {
        Key => 'Description',
        Name => 'Description',
        Searchable => 1,
        Input => {
            Type => 'TextArea',
        },
    },
    {
        Key => 'Type',
        Name => 'Type',
        Searchable => 1,
        Input => {
            Type => 'GeneralCatalog',
            Class => 'ITSM::ConfigItem::Computer::Type',
            Translation => 1,
        },
    },
    {
        Key => 'Owner',
        Name => 'User',
        Searchable => 1,
        Input => {
            Type => 'Customer',
        },
    },
    {
        Key => 'CPU',
        Name => 'CPU',
        InventoryContentXPath => 'HW/Central Processor\(s\)/CPU Brand Name',
        Input => {
            Type => 'Text',
            Size => 50,
            MaxLength => 100,
        },
        CountMax => 16,
        Sub => [
            {
```

```
                Key => 'Cores',
                Name => 'Kerne',
                InventoryContentXPath => 'HW/Central Processor\(s\)/Number of CPU
Cores',
                Input => {
                    Type => 'Text',
                    Size => 20,
                    MaxLength => 10,
                },
            },
        ],
    },
    {
        Key => 'Ram',
        Name => 'Ram',
        Input => {
            Type => 'Dummy',
        },
        Sub => [
            {
                Key => 'MemoryBank',
                Name => 'Bank',
                InventoryContentXPath => 'HW/Memory/Row Size',
                Input => {
                    Type => 'Text',
                    Size => 50,
                    MaxLength => 100,
                },
                CountMax => 16,
            },
        ],
    },
    {
        Key => 'Drives',
        Name => 'Drives',
        InventoryContentXPath => 'HW/Drives::IDE Drives/Device Type',
        InventoryContentEvalString => '$Inventory{\'Device Type\'} eq \'Disk drive\'',
        Input => {
            Type => 'Text',
            Size => 50,
            MaxLength => 100,
        },
        CountMax => 10,
        Sub => [
            {
                Key => 'Model',
                Name => 'Model',
                InventoryContentXPath => 'HW/Drives::IDE Drives/Drive Model',
                Input => {
                    Type => 'Text',
                    Size => 50,
                    MaxLength => 50,
                },
```

```
            },
        ],
    },
    {
        Key => 'CD-Rom',
        Name => 'CD-Rom',
        Searchable => 1,
        InventoryContentXPath => 'HW/Drives::IDE Drives/Drive Model',
        InventoryContentEvalString => '$Inventory{\'Drive can read\'}',
        Input => {
            Type => 'Text',
            Size => 50,
            MaxLength => 100,
        },
    },
    {
        Key => 'FQDN',
        Name => 'FQDN',
        Searchable => 1,
        Input => {
            Type => 'Text',
            Size => 50,
            MaxLength => 100,
        },
    },
    {
        Key => 'NIC',
        Name => 'Network Adapter',
        InventoryContentXPath => 'HW/Network/Network Card',
        Input => {
            Type => 'Text',
            Size => 50,
            MaxLength => 100,
        },
        CountMin => 0,
        CountMax => 10,
        CountDefault => 1,
        Sub => [
            {
                Key => 'IPoverDHCP',
                Name => 'IP over DHCP',
                Input => {
                    Type => 'GeneralCatalog',
                    Class => 'ITSM::ConfigItem::YesNo',
                    Translation => 1,
                },
            },
            {
                Key => 'IPAddress',
                Name => 'IP Address',
                Searchable => 1,
                Input => {
                    Type => 'Text',
```

```
                Size => 40,
                MaxLength => 40,
            },
            CountMin => 0,
            CountMax => 20,
            CountDefault => 0,
        },
    ],
},
{
    Key => 'GraphicAdapter',
    Name => 'Graphic Adapter',
    InventoryContentXPath => 'HW/Video Adapter/Video Chipset',
    InventoryContentEvalString => '$Inventory{\'Video Chipset\'}',
    Input => {
        Type => 'Text',
        Size => 50,
        MaxLength => 100,
    },
},
{
    Key => 'Software',
    Name => 'Software',
    InventoryContentXPath => 'SW/.*?/Name',
    InventoryContentEvalString => '$Inventory{Name}',
    Input => {
        Type => 'Text',
        Size => 50,
        MaxLength => 100,
    },
    CountMax => 1000,
    Sub => [
        {
            Key => 'Version',
            Name => 'Version',
            InventoryContentXPath => 'SW/.*?/Version',
            Input => {
                Type => 'Text',
                Size => 20,
                MaxLength => 20,
            },
        },
        {
            Key => 'SoftwareVendor',
            Name => 'Vendor',
            InventoryContentXPath => 'SW/.*?/Producer',
            Input => {
                Type => 'Text',
                Size => 50,
                MaxLength => 255,
            },
        },
    ],
```

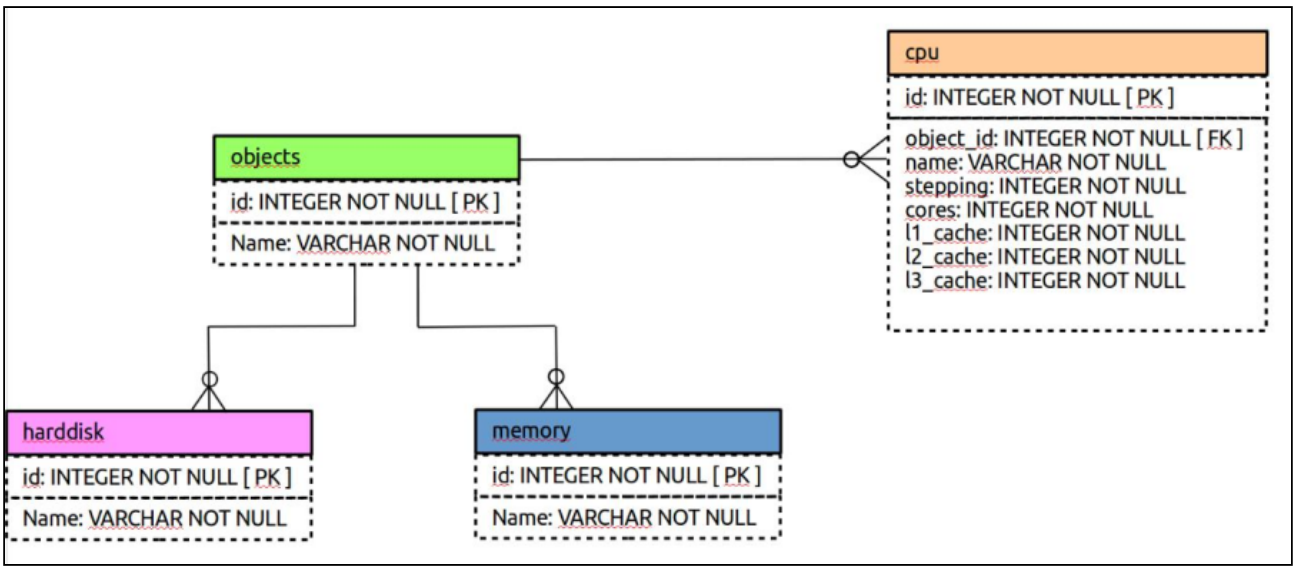
```
    },
    {
        Key => 'OtherEquipment',
        Name => 'Other Equipment',
        Input => {
            Type => 'TextArea',
        },
        CountMin => 0,
        CountDefault => 0,
    },
    {
        Key => 'WarrantyExpirationDate',
        Name => 'Warranty Expiration Date',
        Searchable => 1,
        Input => {
            Type => 'Date',
        },
    },
    {
        Key => 'InstallDate',
        Name => 'Install Date',
        Searchable => 1,
        Input => {
            Type => 'Date',
        },
        CountMin => 0,
        CountDefault => 0,
    },
    {
        Key => 'Note',
        Name => 'Note',
        Searchable => 1,
        Input => {
            Type => 'TextArea',
        },
        CountMin => 0,
        CountDefault => 0,
    },
];
```

4 Inventory Add-On Module, DB Backend

4.1 DB Backend Installation

4.1.1 Prerequisites

- KIX Professional 17 – current version
- An external database with inventory data, organized in a table matrix; see following figure below



This figure is just a schematic example to show you how the data has to be organized to work with this "DB" backend. You can have as many tables with as many attributes as you like. But you must always have a foreign key for the table with all objects (e.g. Server, Mobile Devices, etc.). This so-called "object table" is the primary object list for InventorySync and may only have one row per object. All other tables contain the specific object data.

4.1.2 Installation

Install the package "kixpro-inventorysync-db" using "Paketverwaltung (Package Administration)" (Menu "Admin" area "System Administration") or using the command line.

4.2 DB Backend Configuration

The package contains an example configuration that you can adapt to your needs. To configure settings for this function, select "KIX Professional" group in SysConfig. After page has reloaded, select subgroup "ITSMConfigItem::InventorySync".

- Sources (see page 49)
- Source – Backend – Mapping (see page 49)
- Configuration in Config.pm (see page 49)
- Inventory Content XPath (see page 51)

4.2.1 Sources

- SysConfig setting: ITSMConfigItem::InventorySync###Sources

A source requires a named communication point. A hash is used for configuration. The hash key is the internal name of the source, for example "opsisrv1". The hash value is the display name for this source, for example "inventory1".

You can add several sources for each external inventory server.

4.2.2 Source – Backend – Mapping

- SysConfig setting: ITSMConfigItem::InventorySync###Backend

This setting defines the backend to be used. A hash is used for configuration. The hash key is the identifier of the source (in our example "inventory1"). The hash value must be specified as "DB" so that the InventorySync module uses the opsi backend for this data source.

4.2.3 Configuration in Config.pm

Only the name (and the internal key) of a DB data source and its mapping to the DBBackend is configured in SysConfig. All other "complex" configurations for the configured DB data sources are carried out in Config.pm in a manner that is familiar from the configuration of CustomerUser backends.

You have to create a configuration hash in Config.pm for each DB data source that you have configured in SysConfig for InventorySync. Here is an example of a configuration for our example data source.

Here, ConfigItem class "Computer" is configured, which receives its data from another MySQL database "ocsweb" on localhost. All objects (Computer) are uniquely contained in the database table (or View) "all_clients", and the identifier for each client is the attribute "HARDWARE_ID" in this table.

The name of the object is contained in the table attribute "NAME". The attribute named as the object identifier (here "HARDWARE_ID") must be present in all other tables/views from which the DB backend is to obtain its data (foreign key).

Common tables for software (SW) and hardware (HW) information are configured like two separate/special tables/views that should be used to obtain all the data relevant for the device types "PROZESSOR (PROCESSOR)" and "CONTROLLER".

In the section *"Mapping"*, the attributes from the configured tables/views are mapped to the data structure that is created for each object. The structure for the attribute *"InventoryContentXPath"* is therefore defined in the ConfigItem classes.

```
$Self - > {
    InventorySyncDB
  } - > {
    inventory1
  } = {
    Params => {
      #
      if you want to use an external database, add the# required settings# DSN =>
'DBI:odbc:yourdsn',
        DSN => 'DBI:mysql:database=ocsweb;host=localhost',
        User => 'Test',
        Password => 'xxxxx',
        SourceCharset => 'utf-8',
        DestCharset => 'utf-8',
    },
    Classes => {
      Computer => {
        Params => {
          #
          client list and key value
          ObjectTable => 'all_clients',
            ObjectID => 'HARDWARE_ID',
            ObjectName => 'NAME', #common SW and HW tables / views# where our
hardware / software info comes from
          HWTable => 'csweb.test_view2',
            SWTable => 'ocsweb.test_view1', #special tables to use
          for PROCESSOR / CONTROLLER mapping
          SpecialTables => {
            HW => {
              PROCESSOR => 'cpu_table',
                CONTROLLER => 'controllers',
            }
          }
        }, #the following mapping defines the structure
        for# InventoryContentXpath
        Mapping => {
          HW => {
              PROCESSOR => {
```

```
                name => 'NAME',
                  type => 'TYPE',
                  version => 'VERSION',
              },
              serialNumber => {
                serialnumber => 'SNR',
              },
              BIOS => {
                description => 'DESCRIPTION',
                  name => 'NAME',
              },
              CONTROLLER => {
                description => 'DESCRIPTION',
                  name => 'NAME',
                  type => 'CONTROLLER_TYPE',
              },
          },
          SW => {
            Software => {
              name => 'NAME',
                version => 'VERSION',
                descriptions => 'DESCRIPTION',
            },
          }
      }
    }
  }
```

4.2.4 Inventory Content XPath

The structure and content of the inventory data (hardware and software) returned from the DB backend is defined in "Config.pm[1]". The XPath generally has the following structure:

- Hardware Xpath:

  HW/<some device key>/<attribute>

- Software Xpath:

  SW/<configured key>/<attribute>

1 http://Config.pm

5 Inventory Add-On Module, OPSI Backend

5.1 OPSI Backend Installation

5.1.1 Prerequisites

- KIX Professional 17 – current version
- An OPSI server

5.1.2 Package Installation

Install the package "kixpro-inventorysync-opsi" using "Paketverwaltung (Package Administration)" (Menu "Admin" area "System Administration") or using the command line.

5.2 OPSI Backend Configuration

The package contains an example configuration that you can adapt to your needs. To configure settings for this function, select "KIX Professional" group in SysConfig. After page has reloaded, select subgroup "ITSMConfigItem::InventorySync".

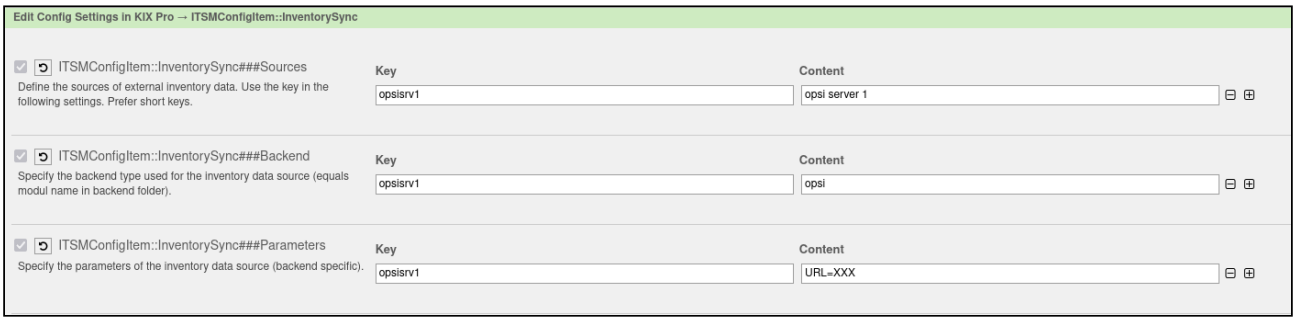


Fig.: Excerpt from SysConfig: KIX Pro → ITSMConfigItem::InventorySync

5.2.1 Source

- SysConfig setting: ITSMConfigItem::InventorySync###Sources

A source requires a named communication point. A hash is used for configuration. The hash key is the internal name of the source, for example "opsisrv1". The hash value is the display name for this source, for example "opsi server 1".
You can add several sources for each external inventory server.

5.2.2 Source Backend Mapping

- SysConfig setting: ITSMConfigItem::InventorySync###Backend

This setting defines the backend to be used. A hash is used for configuration. The hash key is the identifier of the source (in our example "opsisrv1"). The hash value must be specified as "opsi" so that the InventorySync module uses the opsi backend for this data source.

5.2.3 Source Parameters

- SysConfig setting: ITSMConfigItem::InventorySync###Parameters

In this setting, you can configure the parameters for using the opsi backend. All parameters must be specified in the following form:

<parameter>=<value>, <parameter>=<value>, ...

Parameter	Benötigt	Beschreibung
URL	x	The URL to the rpc interface of the opsi server (usually http://<host>:4447/rpc)
User	x	User name of the database user.
Password	x	Password of the database user.

5.2.4 SSL Verification

- SysConfig subgroup *"ITSMConfigItem::InventorySync::opsi*
- SysConfig setting: *InventorySync::Backend::opsi###UserAgentVerifyHostname*

Using this setting, you can switch on/off the SSL certificate check for the configured host.

5.2.5 InventoryContentXPath Structure

The structure and content of the inventory data returned from opsi (hardware and software) is defined on the opsi server under */etc/opsi/hwaudit*.

The Xpath for opsi has the following structure:

- Hardware XPath:

  HW/<opsi device class>/<attribute>

- Software XPath:

  SW/<software registration key pattern>/<attribute>

Typical XPath for opsi may be:

- HW/PROCESSOR/name , to obtain the name attribute of each CPU
- HW/HARDDISK_DRIVE/model, to obtain the model attribute for each hard disk drive

5.3 Example of ConfigItem Class

The following example shows a ConfigItem class definition for the class "Computer" with some changes for the opsi backend based on standard hardware scans. Please note that these are not required changes. You can make your own changes. You must just ensure that the necessary ConfigItem attributes receive values during synchronization. Otherwise an error message will be displayed.

```
[
    {
        Key => 'Vendor',
        Name => 'Vendor',
        Searchable => 1,
        Input => {
            Type => 'Text',
            Size => 50,
            MaxLength => 50,
        },
    },
    {
        Key => 'Model',
        Name => 'Model',
        InventoryContentXPath => 'HW/COMPUTER_SYSTEM/model',
        Searchable => 1,
        Input => {
            Type => 'Text',
            Size => 50,
            MaxLength => 50,
        },
    },
    {
        Key => 'Description',
        Name => 'Description',
        Searchable => 1,
        Input => {
            Type => 'TextArea',
        },
    },
    {
        Key => 'Type',
        Name => 'Type',
        Searchable => 1,
        Input => {
            Type => 'GeneralCatalog',
            Class => 'ITSM::ConfigItem::Computer::Type',
            Translation => 1,
        },
    },
    {
        Key => 'Owner',
```

```
        Name => 'Owner',
        Searchable => 1,
        Input => {
            Type => 'Customer',
        },
    },
    {
        Key => 'SerialNumber',
        Name => 'Serial Number',
        InventoryContentXPath => 'HW/BASE_BOARD/serialnumber',
        Searchable => 1,
        Input => {
            Type => 'Text',
            Size => 50,
            MaxLength => 100,
        },
    },
    {
        Key => 'Software',
        Name => 'Software',
        InventoryContentXPath => 'SW/.*?/displayName',
        InventoryContentEvalString => '$Inventory{displayVersion}',
        Input => {
            Type => 'Text',
            Size => 50,
            MaxLength => 100,
        },
        CountMax => 1000,
        Sub =>   [
            {
                Key => 'Version',
                Name => 'Version',
                InventoryContentXPath => 'SW/.*?/displayVersion',
                Input => {
                    Type => 'Text',
                    Size => 20,
                    MaxLength => 20,
                },
            },
        ],
    },
    {
        Key => 'CPU',
        Name => 'CPU',
        InventoryContentXPath => 'HW/PROCESSOR/name',
        Input => {
            Type => 'Text',
            Size => 50,
            MaxLength => 100,
        },
        CountMax => 16,
    },
    {
```

```
        Key => 'Ram',
        Name => 'Ram',
        InventoryContentXPath => 'HW/MEMORY_MODULE/memorytype',
        Input => {
            Type => 'Text',
            Size => 50,
            MaxLength => 100,
        },
        CountMax => 10,
    },
    {
        Key => 'HardDisk',
        Name => 'Hard Disk',
        InventoryContentXPath => 'HW/HARDDISK_DRIVE/name',
        Input => {
            Type => 'Text',
            Size => 50,
            MaxLength => 100,
        },
        CountMax => 10,
        Sub =>   [
            {
                Key => 'Capacity',
                Name => 'Capacity',
                InventoryContentXPath => 'HW/HARDDISK_DRIVE/size',
                Input => {
                    Type => 'Text',
                    Size => 20,
                },
            },
        ],
    },
    {
        Key => 'CD-Rom',
        Name => 'CD-Rom',
        Searchable => 1,
        InventoryContentXPath => 'HW/OPTICAL_DRIVE/name',
        Input => {
            Type => 'Text',
            Size => 50,
            MaxLength => 100,
        },
    },
    {
        Key => 'FQDN',
        Name => 'FQDN',
        Searchable => 1,
        InventoryContentXPath => 'HW/COMPUTER_SYSTEM/name',
        Input => {
            Type => 'Text',
            Size => 50,
            MaxLength => 100,
        },
```

```
    },
    {
        Key => 'NIC',
        Name => 'Network Adapter',
        InventoryContentXPath => 'HW/NETWORK_CONTROLLER/name',
        InventoryContentEvalString => '$Inventory{ipAddress}',
        Input => {
            Type => 'Text',
            Size => 50,
            MaxLength => 100,
        },
        CountMin => 0,
        CountMax => 10,
        CountDefault => 1,
        Sub =>   [
            {
                Key => 'IPoverDHCP',
                Name => 'IP over DHCP',
                Input => {
                    Type => 'GeneralCatalog',
                    Class => 'ITSM::ConfigItem::YesNo',
                    Translation => 1,
                },
            },
            {
                Key => 'IPAddress',
                Name => 'IP Address',
                Searchable => 1,
                InventoryContentXPath=> 'HW/NETWORK_CONTROLLER/ipAddress',
                Input => {
                    Type => 'Text',
                    Size => 40,
                    MaxLength => 40,
                },
                CountMin => 0,
                CountMax => 20,
                CountDefault => 0,
            },
        ],
    },
    {
        Key => 'GraphicAdapter',
        Name => 'Graphic Adapter',
        InventoryContentXPath => 'HW/VIDEO_CONTROLLER/name',
        InventoryContentEvalString => '$Inventory{deviceId}',
        Input => {
            Type => 'Text',
            Size => 50,
            MaxLength => 100,
        },
    },
    {
        Key => 'OtherEquipment',
```

```
        Name => 'Other Equipment',
        Input => {
            Type => 'TextArea',
        },
        CountMin => 0,
        CountDefault => 0
     },
     {
        Key => 'WarrantyExpirationDate',
        Name => 'Warranty Expiration Date',
        Searchable => 1,
        Input => {
            Type => 'Date',
        },
    },
    {
        Key => 'InstallDate',
        Name => 'Install Date',
        Searchable => 1,
        Input => {
            Type => 'Date',
        },
        CountMin => 0,
        CountDefault => 0,
    },
    {
        Key => 'Note',
        Name => 'Note',
        Searchable => 1,
        Input => {
            Type => 'TextArea',
        },
        CountMin => 0,
        CountDefault => 0,
    },
];
```

6 Inventory Add-On Module, VMWare Backend

6.1 VMWare Backend Installation

6.1.1 Prerequisites

- KIX Professional 17 – current version
- Installed vSphere Perl SDK for vSphere 5.5+ on the KIX server
- A vCenter user with read-only permissions (a user with the standard "read-only" role assigned in datacenter level is usually sufficient)

6.1.2 Package Installation

Install the package "kixpro-inventorysync-vmware" using "Paketverwaltung (Package Administration)" (Menu "Admin" area "System Administration") or using the command line.

6.2 VMWare Backend Configuration

- Basics (see page 63)
- SysConfig Options (see page 64)
- Source (see page 64)
- Source Backend Mapping (see page 64)
- Source Parameters (see page 64)
- Alternative ObjectType Mapping (see page 65)
- ObjectType Fallback (see page 65)
- SSL Verification (see page 65)
- InventoryContentXPath Structure (see page 66)
- Example of ConfigItem Classes for VMWare Backend (see page 81)

6.2.1 Basics

The VMWare backend works on the basis of object types. Each object type has its own set of attributes and provides the data that is mapped to the ConfigItem class attributes in the ConfigItem class definition.

It is always necessary to assign an object type, either at the level of the data source itself, or at the level of the ConfigItem class name mapping, or on the basis of the name of the ConfigItem class itself.

The following object types are possible:

- Datacenter
- Datastore
- Cluster
- Host
- VM

Example:

If you have a ConfigItem class with the name "VMWare-Servers", you have to assign the object type because this provides the data for a VMWare host system.

- Option 1: At the level of the data source

Define a data source and add ";ObjectType=Host" to the end of the parameter line in SysConfig setting: *ITSMConfigItem::InventorySync###Parameters.*

- Option 2: Define a class name mapping

Create a ConfigItem class mapping, where the key is "VMWare-Servers" and the value defines the object type.

SysConfig setting: *InventorySync::Backend::VMWare###CIClassObjectTypeMapping* .

- Option 2: Rename the ConfigItem class

If you do not want to use option 1 or 2, there is still the possibility to rename the ConfigItem class of "VMWare-Servers" in "Host". In this case, the name of the ConfigItem class represents the object type to be used.

6.2.2 SysConfig Options

To configure settings for this function, select "KIX Professional" group in SysConfig. After page has reloaded, select subgroup "ITSMConfigItem:: InventorySync".

6.2.3 Source

- SysConfig setting: ITSMConfigItem::InventorySync###Sources

A source requires a named communication point. A hash is used for configuration. The hash key is the internal name of the source, for example "VMWare". The hash value is the display name for this source, for example "my VMWare environment". You can add several sources for each external inventory server.

6.2.4 Source Backend Mapping

- SysConfig setting: ITSMConfigItem::InventorySync###Backend

This setting defines the backend to be used. A hash is used for configuration. The hash key is the identifier of the source (in our example "VMWare"). The hash value must be specified as "VMWare" so that the InventorySync module uses the VMWare backend for this data source.

6.2.5 Source Parameters

- SysConfig setting: ITSMConfigItem::InventorySync###Parameters

In this setting, you can configure the parameters for using the VMWare backend. All parameters must be specified in the following form:

<parameter>=<value>;<value>;...

Parameter	**Required**	**Description**
URL	x	The URL for the web service interface of the vCenter server (usually http://<host>:443/sdk/webService)

User	x	User name of the database user.
Password	x	Password of the database user.
ObjectType		Optional parameter to restrict the data source as regards a specific object type. Possible values are: • Datacenter • Datastore • Cluster • Host • VM

6.2.6 Alternative ObjectType Mapping

- SysConfig setting: *InventorySync::Backend::VMWare###CIClassObjectTypeMapping*

You can find this setting in the subgroup "ITSM::ConfigItem::InventorySync::VMWare". If, in the data source itself, no restrictions regarding a certain object type are configured, you can assign the object type to be used for each ConfigItem class here. The hash key represents the name of the ConfigItem class and the value represents the object type. Following object types are possible:

- *Datacenter*
- *Datastore*
- *Cluster*
- *Host*
- *VM*

6.2.7 ObjectType Fallback

If the data source has not been restricted to a certain object type, and no class-based mapping exists, the system uses the name of the CI class to determine the object type.

6.2.8 SSL Verification

- SysConfig subgroup *"ITSMConfigItem::InventorySync::opsi*
- SysConfig setting: *InventorySync::Backend::VMWare###UserAgentVerifyHostname*

Using this setting, you can switch on/off the SSL certificate check for the configured host.

6.2.9 InventoryContentXPath Structure

The expansion of ConfigItem classes and the use of InventoryContentXPath parameters are described in Chapter *"KIX SysConfig/KIX Professional/InventorySync"*.

The Xpath for VMWare has the following structure:

- Hardware XPath: HW/<Container>/<Attribute>
- Software XPath: No software XPath currently exists

The attributes and containers available depend on the permissions of the user that you use to connect to the vCenter server. Its also depends on the state of the corresponding objects. Not all of these attributes are always available.

ObjectType	Container	Attribute
Datacenter	Info	configStatus
		numVMDKs
		numHosts
		numClusters
		numDatastores
		numVMs
		overallStatus
	Clusters	name
	Datastores	name
	Host	name
	Network	name

ObjectType	Container	Attribute
Datastore	Info	LUN
		build
		capacity
		capacityGB
		capacityKB
		capacityMB
		freeSpace
		freeSpaceGB
		freeSpaceKB
		freeSpaceMB
		instanceUuid
		isSSD
		numHosts
		numVMs
		type
		url
		version
		vmfsVersion

ObjectType	Container	Attribute
	Hosts	name
	VMs	name

ObjectType	Container	Attribute
Cluster	Info	actionHistory

ObjectType	Container	Attribute
		alarmActionsEnabled
		apiType
		apiVersion
		build
		configStatus
		disableMethod
		effectiveCpuGhz
		effectiveCpuMhz
		effectiveMemory
		effectiveMemory
		effectiveMemoryGB
		effectiveMemoryKB
		effectiveMemoryMB
		fullName
		hBDatastoreCandidatePolicy
		instanceUuid
		licenseProductName
		licenseProductVersion

ObjectType	Container	Attribute
		localeBuild
		localeVersion
		name
		numCpuCores
		numCpuThreads
		numDatastores
		numEffectiveHosts
		numHosts
		numVMs
		osType
		overallStatus
		productLineId
		resourcePool
		totalCpuGhz
		totalCpuMhz
		totalMemory
		totalMemoryKB
		totalMemoryMB

ObjectType	Container	Attribute
		totalMemoryGB
		vendor
		version
	HA	admissionControl
		configuredFailoverCapacity
		dasConfig
		dasIsolationAddress
		defaultVmSettings.dasIsolationResponse
		defaultVmSettings.restartPriority
		defaultVmSettings.vmToolsMonitoringSettings
		defaultVmSettings.vmToolsMonitoringSettings.clusterSettings
		defaultVmSettings.vmToolsMonitoringSettings.failureInterval
		defaultVmSettings.vmToolsMonitoringSettings.maxFailureWindow
		defaultVmSettings.vmToolsMonitoringSettings.maxFailures
		defaultVmSettings.vmToolsMonitoringSettings.minUpTime

ObjectType	Container	Attribute
		defaultVmSettings.vmToolsMonitoringSettings.vmMonitoring
		failover.Level
		hostMonitoring
		status
		vmMonitoring
	DRS	defaultVmBehavior
		status
	Datastores	extent.diskName
		extent.partition
		name
		url
	Hosts	name
	VMs	name
Hosts	Info	bootTime
		build
		cluster
		datacenter

ObjectType	Container	Attribute
		DNS.dhcp
		DNS.hostname
		DNS.nameserver
		DNS.searchDomains
		defaultGateway
		cpuMhz
		cpuGhz
		cpuModel
		datacenter
		fullname
		hostMaxVirtualDiskCapacity
		hostMaxVirtualDiskCapacityKB
		hostMaxVirtualDiskCapacityMB
		hostMaxVirtualDiskCapacityGB
		licenseProductName
		memorySizeGB
		memorySizeKB
		memorySizeMB

ObjectType	Container	Attribute
		model
		name
		numCpuCores
		num CpuPkgs
		numCpuThreads
		numDatastores
		numHBAs
		numNICs
		numVMs
		port
		powerState
		uptime
		uuid
		vendor
		version
		vmotion.ipAdress
		vmotion.subnetMask

ObjectType	Container	Attribute
	vSwitches	mtu
		name
		numPorts
		numPortsAvailable
		portgroup.name[2]
		portgroup.usedPorts
		portgroup.vlanId
		usedPorts
	NICs	autoNegotiateSupported
		dhcp
		driver
		duplex
		ipAdress
		macAdress
		name
		pci
		resourcePoolSchedulerAllowed

2 http://portgroup.name

ObjectType	Container	Attribute
		speedMb
		subnetMask
		wakeOnLanSupported
	VMs	name
VM	Info	cluster
		faultToleranceState
		guestId
		guestMemoryUsage
		guestMemoryUsageKB
		guestMemoryUsageMB
		guestMemoryUsageGB
		guestOS
		host
		hostMemoryUsage
		hostMemoryUsageKB
		hostMemoryUsageMB
		hostMemoryUsageGB
		host

ObjectType	Container	Attribute
		instanceUuid
		ipAddress
		isTemplate
		memorySize
		memorySizeKB
		memorySizeMB
		memorySizeGB
		name
		numCPU
		numEthernetCards
		numVirtualDisks
		overallStatus
		resourcePool
		storageFree
		storageFreeKB
		storageFreeMB
		storageFreeGB
		storageTotal

ObjectType	Container	Attribute
		storageTotalKB
		storageTotalMB
		storageTotalGB
		storageUsed
		storageUsedKB
		storageUsedMB
		storageUsedGB
		toolsStatus
		uptime
		vmPathName
		vmState
	CDRoms	label
		summary
	Datastores	name
	VDisks	backingObjectId
		capacity
		capacityKB
		capacityMB

ObjectType	Container	Attribute
		capacityGB
		datastore
		diskMode
		filename
		label
		size
		type
		uuid
		writeTrough
	VNICs	ipAddress
		label
		macAdress
		network
	Snapshots	createTime
		description
		name
		quiesced
		state

Please note

The vCenter API also defines Virtual Machine Templates as Virtual Machine Objects. You can see whether a VM is a template or not by looking at the "isTemplate" attribute

6.2.10 Example of ConfigItem Classes for VMWare Backend

The package contains example ConfigItem class definitions for all object types. You can find these examples in the directory:

```
/opt/kix/var/packagesetup/InventorySyncVMWareBackend
```

Note

Please note that these are just examples which contain all available attributes, including some attributes with utilization values (for example, current memory or processor usage) or different units (kB,MB,GB,Mhz, Ghz,...) . You can adapt these definitions in accordance with your wishes.
Important: Please note that mapped utilization values usually create a new version of a ConfigItem because they are subject to frequent changes. You should remove them from your ConfigItem class definition if these values are not of interest in your CMDB.
You must also ensure that essential ConfigItem attributes receive values

These examples use ConfigItem attribute *"CICReference"* that is already contained in KIX. It is very useful to create links between different ConfigItems automatically.

To make the most of the advantages of these configuration examples, you should first of all import and then synchronize so that all objects are already available in the synchronization process, and can therefore be linked.

7 ITSM Standardworkflows

7.1 Installation of ITSM-Standard Workflows Add-On Module

7.1.1 Prerequisites

- KIX Professional 17 – current version

7.1.2 Package Installation

Install the package "KIXITSMTicketStatusWorkflows" using "Paketverwaltung (Package Administration)" (Menu "Admin" area "System Administration") or using the command line.

7.2 Ticket Types

Aim is the mapping of ticket-type-specific state workflows. For each ticket type there are suitable states and possible state transitions – both automatic as well as manual. The basic function of the ticket state is provided by KIX. The applicable configurations have been adapted in the add-on package.

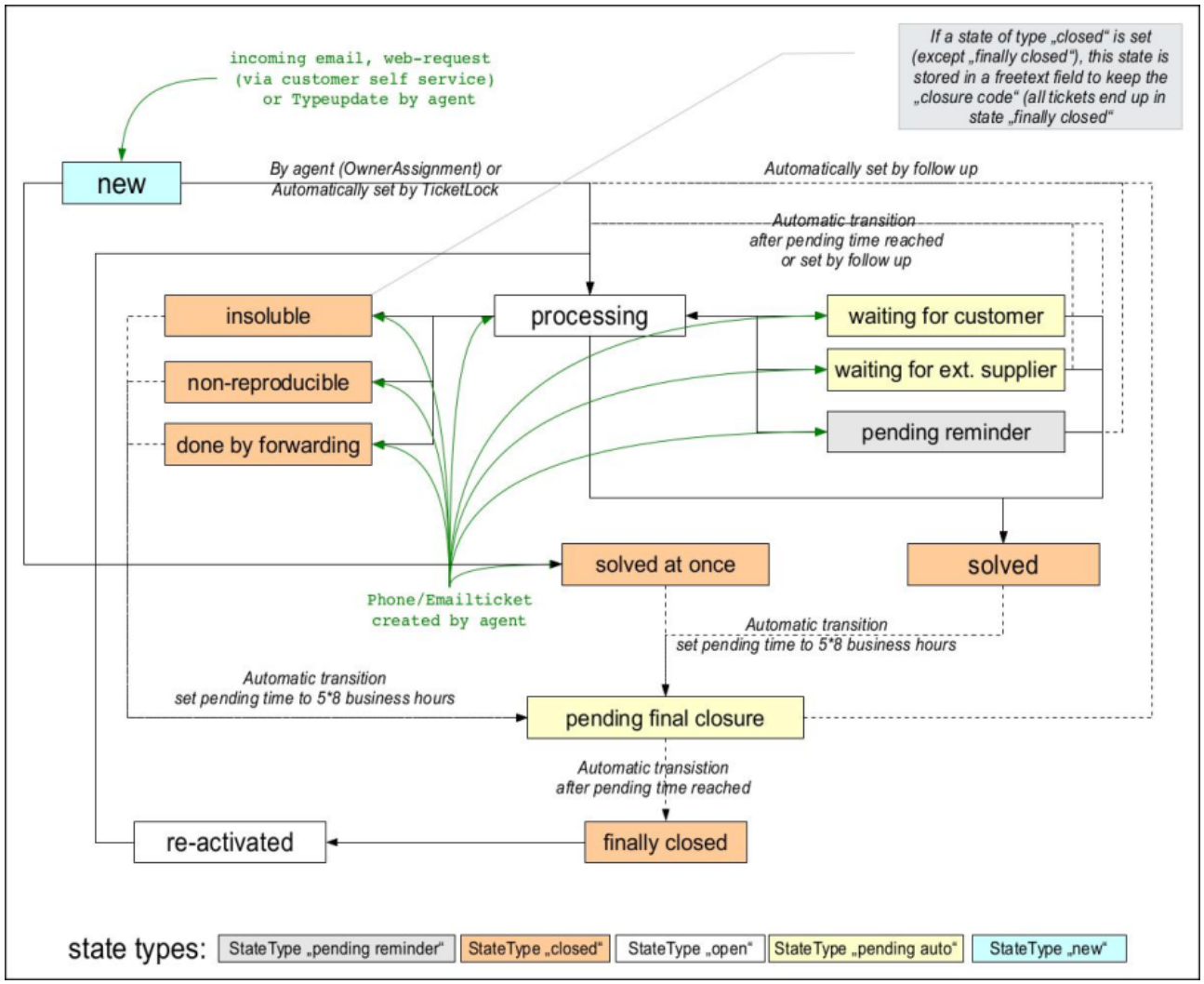
Furthermore, additional TicketStateWorkflow ACLs and ticket events not available in KIX have been created. All ticket states and ticket types not yet available/deactivated are created/activated when the package is installed. All obsolete or irrelevant ticket types are deactivated/renamed in the process and are no longer selectable.

7.2.1 Relevant Ticket States

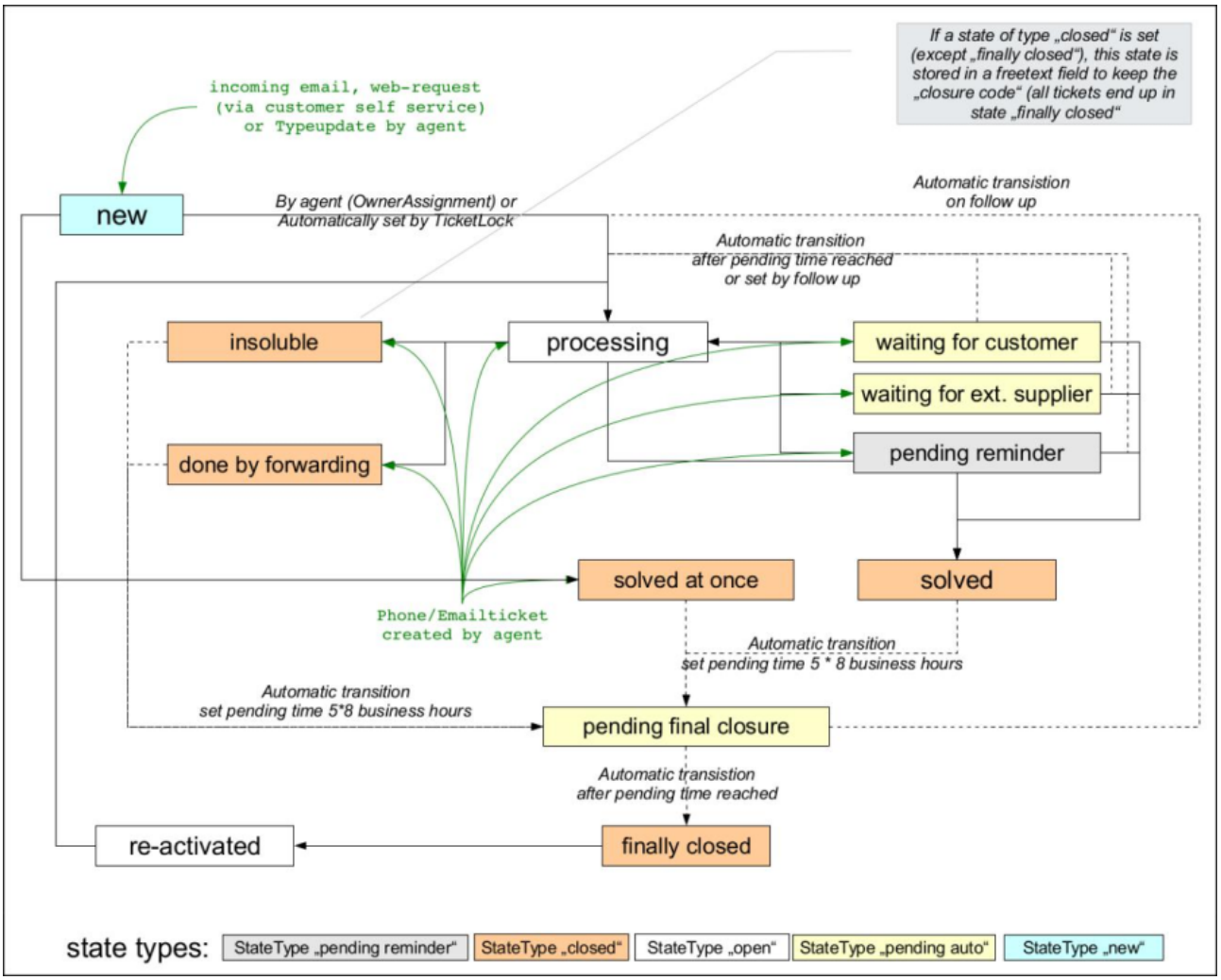
State name	**State type**
closed with workaround	closed
done by forwarding	closed
finally closed	closed
insoluble	closed
non-reproductible	closed
rejected	closed
canceled	closed
solved	closed
solved at once	closed
solved by change implementation	closed
new	new

problem candidate	new
accepted	open
processing	open
re-activated	open
workaround documented	open
review	open
pending final closure	pending auto
waiting for customer	pending auto
waiting for external supplier	pending auto
pending reminder	pendig reminder
pending for change implementation	pending reminder

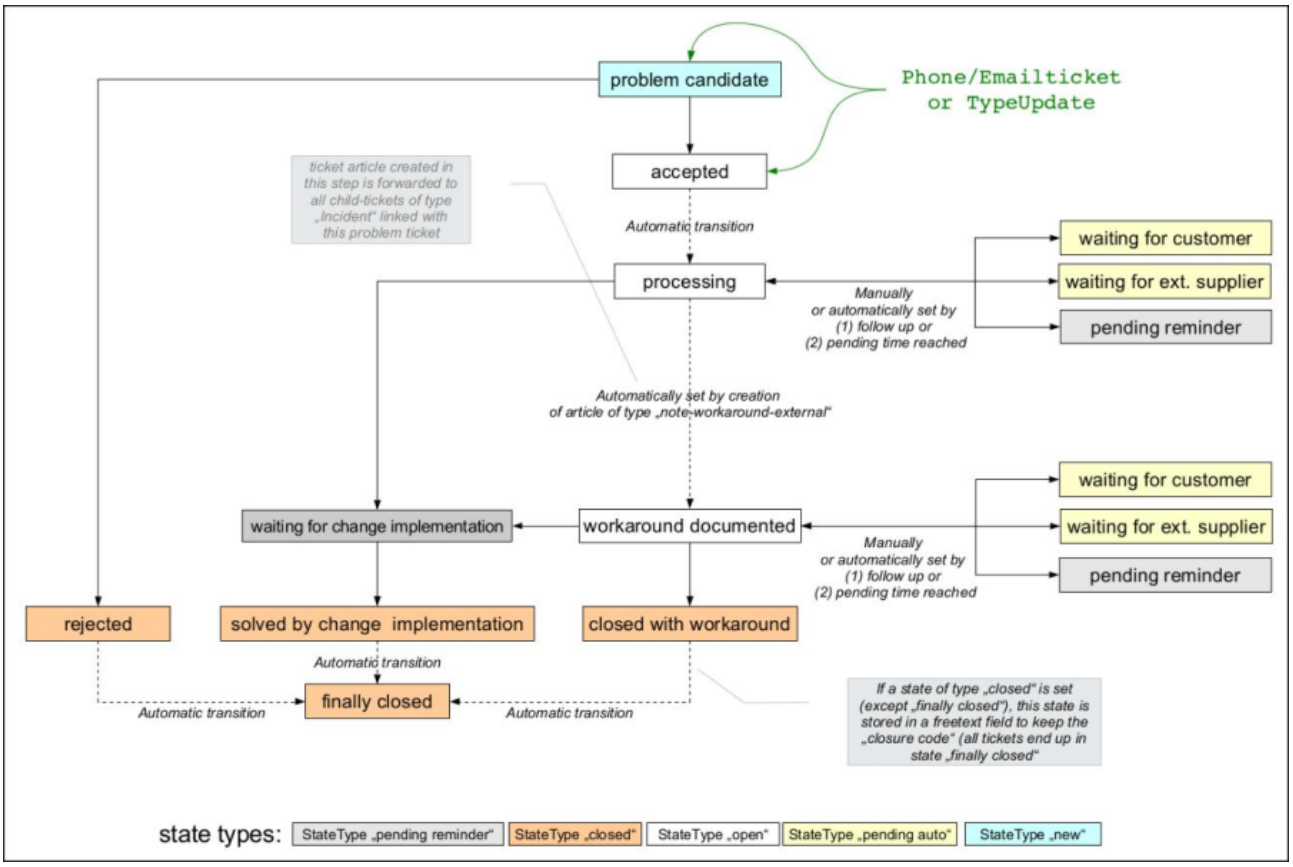
7.2.1.1 Workflow "Incident" + "Incident::Disaster"



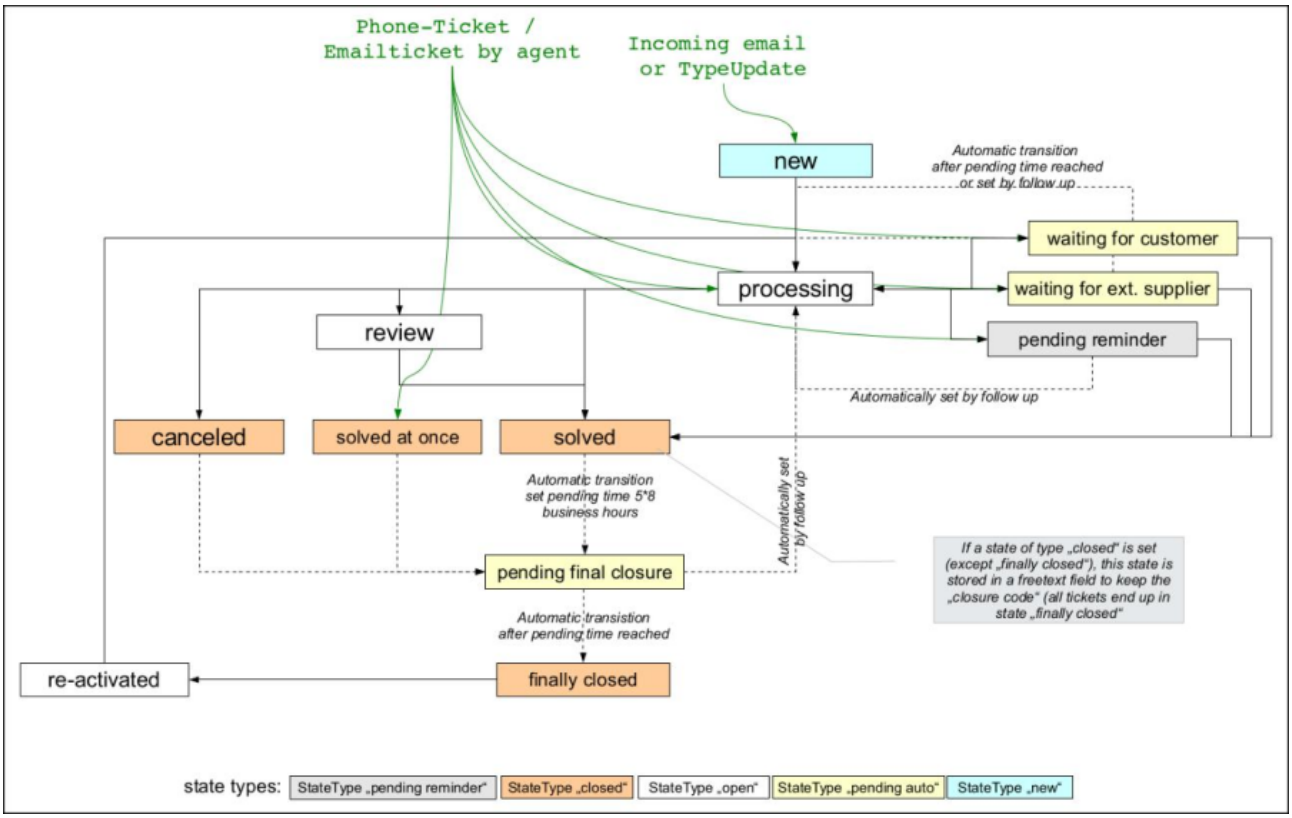
7.2.1.2 Workflow "Service Request" + "Information Request"



7.2.1.3 Workflow "Problem"



7.2.1.4 Workflow "RfC::Standard"



7.2.2 Saving the Closure Code

When a ticket state of the type "closed" ("geschlossen") is reached, this is written as a closure code to a dynamic field. The state "finally closed" is excluded from this automatism. You can configure which dynamic field is used for the application of the closure code.

> **Please note**
>
> The closure code is shown in the ticket view without translation, just as it was saved. If a translation is desired, a so-called CallMethod can be configured for the ticket view in SysConfig.

Example:

Expansion of setting: "*Ticket::Frontend::AgentTicketZoom###TicketDataKeys*"

with "Key" = 2000 and value = "*CallMethod::TicketObject::GetClosureCodeTranslated*"

Expansion of setting: "Ticket::Frontend::AgentTicketZoom###TicketDataLabel"

with "Key" = "CallMethod::TicketObject::GetClosureCodeTranslated" and value = ClosureCode

7.2.3 Ticket Action "Record Workaround"

This ticket action only appears for the ticket type "Problem" in the specific state (see configuration for Match::ExcludedAction::HideWorkaroundNote) and enables an article to be created in the same way as the "Note" function, with the difference in the assigned article type and a separate ticket action. The article types "Workaround (extern) (external)" / "Workaround (intern) (internal)" indicate notes/articles in which a workaround is described.

Furthermore, this article is forwarded to linked child tickets of a configurable ticket type. Here, the forwarded article obtains the type of an internal note. In this way, workarounds that are created in a problem ticket are forwarded to linked fault tickets.

7.2.4 Ticket Action "Close"

This ticket function is only shown when a follow-on state of the type "Geschlossen" (closed) is assigned to the current ticket state. This is defined in the applicable ticket state workflow.

7.3 Configuration of ITSM-Standard Workflows

Key name	Explanation
Ticket::Acl::Module###950_TicketStateWorkflow	Ticket ACLs to define the possible follow-on state.
TicketStateWorkflow	Settings for the ticket state workflow, to define the possible follow-on state. States have to be comma separated. Placeholders are *ANY*, *PREVIOUS*, and *NONE*.
TicketStateWorkflow::FallbackForPreviousState	Ticket state used if "*PREVIOUS*" could not be determined.
TicketStateWorkflow::DefaultTicketState	Settings for DefaultTicketState, to define the possible follow-on state. States have to be comma separated. Placeholders are *ANY*, *PREVIOUS*, and *NONE*.
Ticket::CreateOptions::DefaultTicketType	Vorausgewählter Tickettyp in AgentTIcketPhone und AgentTicketEmail
Ticket::EventModulePost###1-ForceStateChangeOnLock	Ticket Event module for automatically setting a new ticket state after the ticket has been locked. The key is ticket type + ":::" + current state; the content is state after locking.
Ticket::EventModulePost###700-TicketStateWorkflowTypeUpdate	Updates ticket state, if configured or necessary, after a change to the ticket type.
TicketStateWorkflow::PostmasterFollowUpState	State of a ticket for which a follow-up applies (use *PREVIOUS* as a placeholder for the last state in the ticket history before the current one).
Ticket::PendingAutoStateType	Defines possible state types for pending tickets.

Key name	Explanation
TicketStateWorkflow::PostmasterFollowUpCheckCustomerIDFrom	Checks whether the sender of a message is a customer user and whether they have the same customer ID as the one saved on the ticket. If this is the case, the email is considered to be *"email-external"*, otherwise *"email-internal"*. Default behavior if deactivated.
TicketStateWorkflowPostmasterFollowUpCheckAgentFrom	Checks whether sender is saved as agent in the database; if so, sets sender type to *"Agent"*.
Ticket::EventModulePost###701-TicketStateWorkflowAutomaticStateAction	Performs ticket actions following state update – currently restricted to state or queue change.
TicketStateWorkflowAutomaticStateAction###NextStateSet	Enables automatic setting of state when another state is achieved.
TicketStateWorkflowAutomaticStateAction###NextStatePendingOffset	If the automatically set state is of the type *"warten (wait)"*, the time configured here is used as waiting time.
Ticket::StateAfterPending	Defines which state is to be set once the waiting time has been reached.
Ticket::Type	Activates the ticket type features.
Ticket::Frontend::AgentTicket*###StateType	Possible state types for certain ticket actions.
Ticket::EventModulePost###650-TicketWriteClosureCodeToDynamicField	De-/activates event handler to save the closure code in a dynamic field.
TicketWriteClosureCodeToDynamicField###DynamicField	Defines which dynamic field is to be used for saving the closure code (default is *"ClosureCode"*).
TicketWriteClosureCodeToDynamicField###NoClosureStates	Defines which "close" state is not to be considered as the closure code for saving in the dynamic field.

Key name	Explanation
Ticket::EventModulePost###800-ProblemWorkaroundNote	Workflow module which forwards workaround articles and updates to ticket state of problem tickets to linked tickets. Parameters: • Event: triggering ticket event • LinkType: Link type to be followed • LinkDirection: Link direction to be followed • TargetTicketType: Ticket types to which the article should be forwarded • TriggerTicketType: Ticket type for which this action is performed • TriggerArticleType: Article type for which this action is performed • TriggerTicketNewState: State of problem ticket after a workaround has been created
Frontend::Module###AgentTicketNoteWorkaroundExternal	Frontend module registration for ticket action "AgentTicketNoteWorkaroundExternal" in agent interface
Ticket::Frontend::MenuModule### 420-Note-Workaround-External	Display ticket action for a workaround note in ticket preview
Ticket::Frontend::AgentTicketNoteWorkaroundExternal###	All of the configurations necessary for the screen "AgentTicketNoteWorkaroundExternal" such as entry of note field, article type, state update, change of owner, etc. as far as these are necessary. Relevant parameters: • ArticleTypes • Note • State and StateDefault

Key name	Explanation
Ticket::Frontend::AgentTicketNotWorkaroundExternal::KIXSidebarBackend	De-/activation of text modules, customer info, linked persons for screen "AgentTicketNoteWorkaroundExternal"
Ticket::ArticleTypeIcon	Defines which icons are used for certain article types.
Ticket::Acl::Module###803-MatchActionRestrictions	Ticket ACL to deactivate certain ticket actions based on ticket data.
Match::ExcludedAction::HideWorkaroundNote	Defines under which conditions the ticket action "Workaround erfassen (Record Workaround)" is NOT visible. The key contains the ticket properties for which the action should not be possible.
Ticket::Acl::Module###880-HideAgentTicketClose	Ticket ACL to restrict "close" functions on the basis of the current ticket state.

8 Kanban Board

8.1 Installation of Kanban Board Add-On Module

8.1.1 Prerequisites

- KIX Professional 17 – current version

8.1.2 Package Installation

Install the package "Kanban-Board" using "Paketverwaltung (Package Administration)" (Menu "Admin" area "System Administration") or using the command line.

8.1.3 New Groups

During installation, two new groups are created. The "kanbanview" group enables access to the Kanban Board and configuration of the Kanban Board. The Kanban Board can be viewed with read-only rights (ro). The tickets can be moved in the Kanban Board with writing rights (rw).
The group "kanbanview_dispatchSidebar" enables access to the "Tickets to dispatch" widget. The initial user (default: root@localhost) automatically has writing rights for both groups.

8.1.4 New Tables

During installation, two new tables are created. The table "kanban4kix_shared_sets" contains all views which users share with other agents. The table "kanban4kix_subscribed_sets" contains the view subscriptions.

8.2 Configuration of Kanban Board Add-On Module

Once you have installed the package, if necessary, you can set the default values in the Admin area/ SysConfig. Select SysConfig group "Kanban4KIX".

After page has reloaded, select one of the subgroups "Frontend::Agent::KanbanView" or "Frontend::Agent:: AgentKanbanActionCommonPopup".

Default configuration is already contained in the package. You can adapt this in accordance with your wishes. The SysConfig settings are explained in the following sections.

- Widget "Tickets to dispatch" in Kanban Sidebar (see page 98)
- Normalization of Column Widths (see page 98)
- Ticket Attributes in Head Area of a Ticket Card (see page 98)
- Ticket Attributes in Content Area of a Ticket Card (see page 98)
- Dynamic Links for Ticket Attributes (see page 99)
- Dummy Attributes (see page 99)
- Progress Bar for Ticket Attributes (see page 99)
- Meta States as Columns (see page 99)
- Ticket Type Colors (see page 100)
- Ticket-Type-Specific CSS Classes (see page 100)
- Dynamic Fields as Swimlane Attributes (see page 100)
- Number of Tickets in Kanban Board (see page 101)
- Access to "Change" Popup (see page 101)
- Setting the Required Lock (see page 101)
- Showing the "Incident State" of Ticket Service (see page 101)
- Changing Different Ticket Attributes and Dynamic Fields when Moving a Ticket in the Kanban Board (see page 102)
- Next Possible State Types (see page 102)
- Adding a Note when Moving a Ticket in the Kanban Board (see page 103)
- Note Field as Mandatory Field (see page 103)
- Default Values for Note Subject and Note Body (see page 103)
- Size of the Note Input Field (see page 104)
- Default Article Type (see page 104)
- Possible Article Types (see page 104)
- History Type (see page 105)
- History Comment (see page 105)

8.2.1 Widget "Tickets to dispatch" in Kanban Sidebar

- SysConfig setting: Frontend::KanbanSidebarBackend###Dispatch

This setting activates "Tickets to dispatch" widget in the Kanban sidebar. Using the key "Group" you can determine which group a user has to be in to be able to use the widget. To display the tickets to dispatch in the widget, the user must also have rights for the pop-up that appears when moving tickets. (Key: "Frontend::Module###AgentKanbanActionCommonPopup").

> **Please note**
>
> Default value for "Group" is "kanbanview_dispatchSidebar".

8.2.2 Normalization of Column Widths

- SysConfig setting: AgentKanbanView::NormalizeColWidth

This setting determines whether the columns should have the same width.

> **Please note**
>
> This setting is required and default value is "No".

8.2.3 Ticket Attributes in Head Area of a Ticket Card

- SysConfig setting: AgentKanbanView::CardContent::HeadArea

This setting specifies which and in what sequence the ticket attributes are visible in the head area of a ticket in the Kanban Board. The number by "Key" determines the sequence. The value by "Content" is the information to be shown, e.g. "Queues". Dynamic fields must be specified in the style "DynamicField*"_.

8.2.4 Ticket Attributes in Content Area of a Ticket Card

- SysConfig setting: AgentKanbanView::CardContent::ContentArea

This setting specifies which and in what sequence the ticket attributes are visible in the content area of a ticket (in "L" ticket view) in the Kanban Board. The number by "Key" determines the sequence. The value by "Content" is the information to be shown, e.g. "Owner". Dynamic fields must be specified in the style "DynamicField*"_.

8.2.5 Dynamic Links for Ticket Attributes

- SysConfig setting: AgentKanbanView::CardContent::AttributeAsLink

Using this setting, you can define a specific link for the ticket attributes in the head and content area of a ticket card. For example, to open the relevant ticket:

```
"index.pl?Action=AgentTicketZoom;TicketID="
```

<TICKET_*> for ticket attributes

<CONFIG_*> for SysConfig options

<ENV_*> for environment variables.

External links require the protocol http://www.somesite.com for example

8.2.6 Dummy Attributes

- SysConfig setting: AgentKanbanView::CardContent::DummyAttribute

Using this setting, you can define a dummy attribute, for example, for ticket-independent links. "Key" is the dummy identifier and "Content" the visible text on the ticket card. The relevant dummy key must be used in "AgentKanbanView::CardContent::HeadArea" or "AgentKanbanView::CardContent::ContentArea" in order to be visible.

8.2.7 Progress Bar for Ticket Attributes

- SysConfig setting: AgentKanbanView::CardContent::ProgressBar

Using this setting, you can specify additional ticket attributes which are displayed as a progress bar in the content area of a ticket card. The number by "Key" determines the sequence. The value by "Content" is the information to be shown.

8.2.8 Meta States as Columns

- SysConfig setting: AgentKanbanView::TicketMetaStates

Using this setting, you can define the columns and their corresponding states. The states are only default options; they can be replaced by personal settings during Kanban configuration. All states should exist and are separated by commas (without an additional space). A state that is assigned once can only appear in one column, not in several (the first appearance counts).

Please note

This setting is required.

8.2.9 Ticket Type Colors

- SysConfig setting: AgentKanbanView::TicketType::Colors

Using this setting, you can define the colors for the ticket types. This setting overwrites the color declarations from CSS class "AgentKanbanView::TicketType:CSSClass".

Please note

This setting is required.

8.2.10 Ticket-Type-Specific CSS Classes

- SysConfig setting: AgentKanbanView::TicketType::CSSClass

Using this setting, you can configure a CSS class for a specific ticket type. The color declarations are overwritten by "AgentKanbanView:TicketType:Colors".

8.2.11 Dynamic Fields as Swimlane Attributes

- SysConfig setting: AgentKanbanView::DynamicFieldsAsSwimlanes

Using this setting, you can define dynamic fields as further swimlanes (lines) in the Kanban Board. Dynamic fields of the type: "Article" / "Date" / "Time" / "Multiselect", and "Text area" cannot be used.

8.2.12 Number of Tickets in Kanban Board

- SysConfig setting: AgentKanbanView::SearchLimit

This setting defines how many tickets are shown as a maximum in the lists (sidebar and ticket type groups) and as Kanban tickets in the Kanban Board.

> **Please note**
>
> This entry is required. Only numbers are possible as a value.

8.2.13 Access to "Change" Popup

- SysConfig setting: Ticket::Frontend::AgentKanbanActionCommonPopup###Permission

This setting defines the rights required to make changes to the configured ticket attributes if the ticket is moved in the Kanban Board.

> **Please note**
>
> This setting is required.

8.2.14 Setting the Required Lock

- SysConfig setting: Ticket::Frontend::AgentKanbanActionCommonPopup###RequiredLock

This setting defines whether a ticket has to be locked to be able to use the "Change" popup. If the ticket has not yet been locked, it is locked and the agent currently logged in is set as the owner.

> **Please note**
>
> This setting is required.

8.2.15 Showing the "Incident State" of Ticket Service

- SysConfig setting: Ticket::Frontend::AgentKanbanActionCommonPopup###ShowIncidentState

This option determines whether the service incident state is to be displayed during service selection in the "Change pop up".

8.2.16 Changing Different Ticket Attributes and Dynamic Fields when Moving a Ticket in the Kanban Board

The following settings determine whether changes to the following ticket attributes are possible in the "Change" popup when moving a ticket in the Kanban Board:

SysConfig key	Explanation
Ticket::Frontend::AgentKanbanActionCommonPopup###Owner	The owner of the ticket can be changed.
Ticket::Frontend::AgentKanbanActionCommonPopup###OwnerMandatory	Selection of the owner is a mandatory field.
Ticket::Frontend::AgentKanbanActionCommonPopup###TicketType	The ticket type can be changed.
Ticket::Frontend::AgentKanbanActionCommonPopup###Queue	The queue of the ticket can be changed.
Ticket::Frontend::AgentKanbanActionCommonPopup###Service	The ticket service can be changed.
Ticket::Frontend::AgentKanbanActionCommonPopup###Responsible	The person responsible for the ticket can be changed.
Ticket::Frontend::AgentKanbanActionCommonPopup###State	The ticket state can be changed.
Ticket::Frontend::AgentKanbanActionCommonPopup###Priority	The ticket priority can be changed.
Ticket::Frontend::AgentKanbanActionCommonPopup###Title	The ticket title can be changed.

8.2.17 Next Possible State Types

- SysConfig setting: Ticket::Frontend::AgentKanbanActionCommonPopup###StateType

Using this setting, you can define which next possible state types can be selected for a ticket in the "Change" popup.

> **Please note**
>
> This setting is required.

8.2.18 Adding a Note when Moving a Ticket in the Kanban Board

- SysConfig setting: Ticket::Frontend::AgentKanbanActionCommonPopup###Note

This setting defines whether adding a note is possible in the "Change" popup.

8.2.19 Note Field as Mandatory Field

- SysConfig setting: Ticket::Frontend::AgentKanbanActionCommonPopup###NoteMandatory

Using this setting, you define whether the note field in the "Change" popup is a mandatory field.

> **Please note**
>
> Default setting is "No". This setting only has an effect if the setting "Ticket::Frontend::AgentKanbanActionCommonPopup###Note" is set to "Yes".

The setting can be overwritten by the setting "Ticket::Frontend::NeedAccountedTime".

8.2.20 Default Values for Note Subject and Note Body

- SysConfig setting: Ticket::Frontend::AgentKanbanCommonPopup###Subject
- SysConfig setting: Ticket::Frontend::AgentKanbanCommonPopup###Body

These settings define the default values for the subject and body of a note which appears in the "Change" popup when moving a ticket in the Kanban Board.

8.2.21 Size of the Note Input Field

- SysConfig setting: Ticket::Frontend::AgentKanbanCommonPopup::TextAreaNoteCols
- SysConfig setting: Ticket::Frontend::AgentKanbanCommonPopup::TextAreaNoteRows

These two settings define the default size of the note input field. If rich text is deactivated, then the values stand for the number of rows or columns (letters per row). If rich text is activated, then the values are calculated in pixels (columns x 10 and rows x 20).

> **Please note**
>
> These settings are required. Default values are 60 and 10.

8.2.22 Default Article Type

- SysConfig setting: Ticket::Frontend::AgentKanbanActionCommonPopup###ArticleDefault

Here you can define the default type for the article which is created with the "Change" popup when moving a ticket in the Kanban Board.

> **Please note**
>
> Default value is "note-internal".

8.2.23 Possible Article Types

- SysConfig setting: Ticket::Frontend::AgentKanbanActionCommonPopup###ArticleTypes

Using this setting, you can define the available note types for the article which is created with the "Change" popup when moving a ticket in the Kanban Board. If this setting is deactivated, the setting from "Ticket::Frontend:: AgentKanbanActionCommonPopup###ArticleDefault" is used as the article type.

8.2.24 History Type

- SysConfig setting: Ticket::Frontend::AgentKanbanActionCommonPopup###HistoryType

Using this setting, you can define the created history type for the ticket history when a ticket is moved in the Kanban Board.

> **Please note**
>
> Default value is "AddNote".

8.2.25 History Comment

- SysConfig setting: Ticket::Frontend::AgentKanbanActionCommonPopup###HistoryComment

Using this option, you can define the created history comment for the ticket history when a ticket is moved in the Kanban Board.

> **Please note**
>
> Default value is "%%Note".

9 KIX Connect

9.1 Installation of KIX Connect Add-On Module

The "KIX Connect" module provides you with pre-configured web services for data exchange via HTTP(S) with external backends. In doing so, the package uses the functionality of the web services and provides you with defined invokers and operations for communicating with the external backends. From KIX as the leading system (requester), it is therefore possible to use services from an external backend (provider), for example to create a ticket in an external KIX. The package currently offers you the following interfaces:

- KIX2KIX for communicating with an external KIX/OTRS as a backend
- KIX2JIRA for communicating with JIRA as a backend

9.1.1 Prerequisites

- KIX2KIX: requester system: KIX17.4 and above / response system: KIX17.x / OTRS 5.x
- KIX2JIRA: The package was tested with JIRA V7.3.2, JIRA REST API version 2

9.1.2 Installation

Install "KIX Connect" module with your operating system's package manager. The package name is "kixpro-connect".

9.1.3 New Dynamic Fields

The following dynamic fields are created during installation:

Name	Label	Function
KIX2JIRAType	Type of JIRA-Issue	Selection of the JIRA issue type. Displayed in: *"Create New Ticket"* dialog
KIX2JIRAProject	Project of JIRA-Issue	Designation of the JIRA project Displayed in: *"Create New Ticket"* dialog after selecting the JIRA issue
KIX2JIRAKey	Key of JIRA-Issue	Indicates the ID of the JIRA-Issue. Displayed in: is not displayed

Name	Label	Function
KIX2JIRASend	Send to JIRA	Checkbox to transfer articles including attachments to JIRA Displayed in: "New Note" tab in ticket zoom view. (If the ID of the JIRA issue has been saved in the dynamic field "KIX2JIRAKey").
KIX2KIXCreate	Create external ticket	Checkbox to create an external ticket Displayed in: "Create New Ticket" dialog
KIX2KIXKey	Key of external ticket	Indicates the ID of the external ticket. Displayed in: is not displayed
KIX2KIXSend	Send to external ticket	Checkbox to transfer articles including attachments to an external ticket Displayed in: "Add note" dialog / "New note" tab in ticket zoom view. (If the ID of the JIRA issue has been saved in the dynamic field "KIX2KIXKey").

9.1.4 New ACLs

The following ACLs are created during installation:

Name	Function
000_KIXConnect_HideSendFields	Hides the fields IX2JIRASend and KIX2KIXSend in the agent frontend (actions which correspond to RegExp /Agent.*/)
001_KIXConnect_HideKIX2JIRAProject	Hides the field IX2JIRAProject in the agent frontend (actions that correspond to RegExp /Agent.*/).
010_KIXConnect_ShowKIX2JIRASend	Shows KIX2JIRASend when a value is saved in KIX2JIRAKey. The value is entered by the web service.
011_KIXConnect_ShowKIX2KIXSend	Shows KIX2KIXSend when a value is saved in KIX2KIXKey. The value is entered by the web service.

Name	Function
012_KIXConnect_ShowKIX2JIRAProject	Shows KIX2JIRAProject when a value has been selected in KIX2JIRAType.

Please note

During package de-installation, the delivered web services, ACLs, and dynamic fields are also deleted.

9.2 Generic Invoker

When setting up web services, you can use the "generic invoker" to call up invokers based on schedules even without being dependent on KIX objects.

Application scenarios

Application scenarios include:

- Creating SessionIDs or tokens for web services which do not accept username and password authentication in each call-up.
- As the basis for alternative InventorySync
- Bi-directional interface KIX-2-Jira, which means active collection of JIRA issues and creation of tickets

9.2.1 Use

Running possibilities

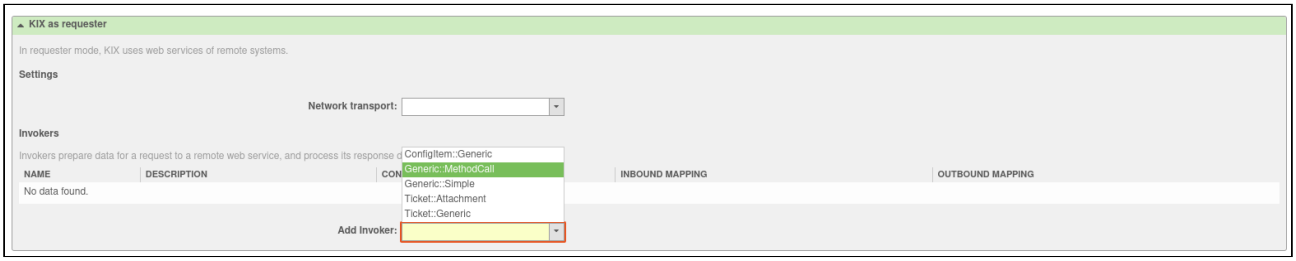
The following running possibilities are available:

- Time-based with schedule configuration as with Generic Agent
- Event-based with access to event data
- Manually in administration of the invoker

> **Note**
>
> Notes: Invoker 'Generic::MethodCall' does not process data itself.

Creating an invoker

Create a new web service and save it. You can then select the generic invoker via "Generic::MethodCall" for "KIX as requester".



General configuration

Prepare

If nothing is configured here, the request is successfully completed at this point. If a method is saved here, the hash key 'Data' is transferred to the configured method of the configured module together with the configured 'PrepareParameters'. The configured 'RunAs' is used as UserID. A hash is expected as the response from the call-up. This is to be used for 'Data' in the request.

Validate

If nothing is configured here, the response from 'Prepare' is forwarded. If a method is saved here, the response from 'Prepare' is transferred to the configured method of the configured module together with the configured 'ValidateParameters'. The configured 'RunAs' is used as UserID. A hash is expected as the response from the call-up. If this is not populated, or if the key 'StopCommunication' is set with a true value, the request is successfully completed at this point.

Response

If nothing is configured here, the data of the response is simply forwarded on. If a method is saved here, the data of the response is transferred to the configured method of the configured module together with the configured 'ResponseParameters'. The configured 'RunAs' is used as UserID. A hash is expected as the response from the call-up. This is to be used for 'Data' in the response.

Static values

If 'Parameter (Statisch)' (Parameters (static)) are configured, these are taken over from the response from 'Prepare', as long as they are defined and are available in the mapping in 'Response'.

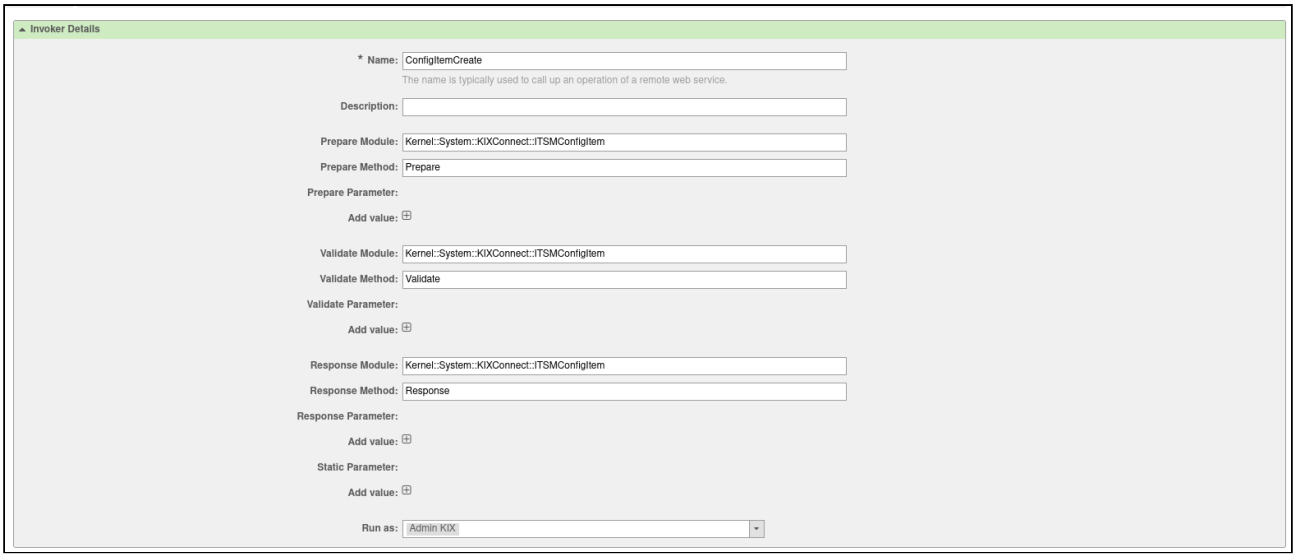


Figure: First Part of Screen for Generic::MethodCall

Mappings

Mappings can be saved on the invoker for the following data:

- Mapping for preparation data
- Mapping for outgoing request data
- Mapping for incoming response data
- Mapping for feedback data

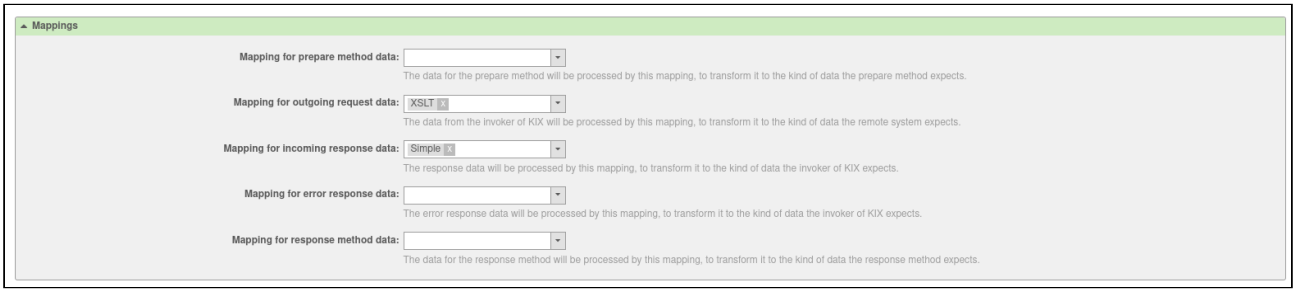


Figure: Mappings for Data

You can choose between the options "Plain", "Ticket template", or "XSLT" for each mapping. Once you have clicked "Save" at the end of the form, a button appears next to the newly selected mapping type that you can use to adapt the mapping, or re-save it.

If you do **not** want to use a mapping, click the small cross on the entry already selected.

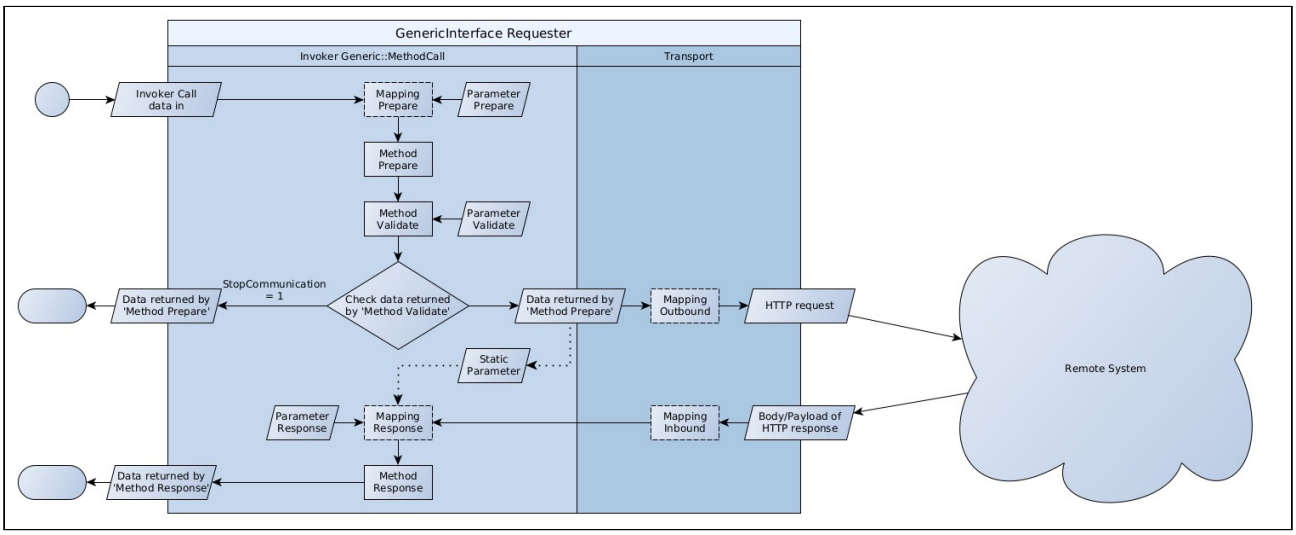


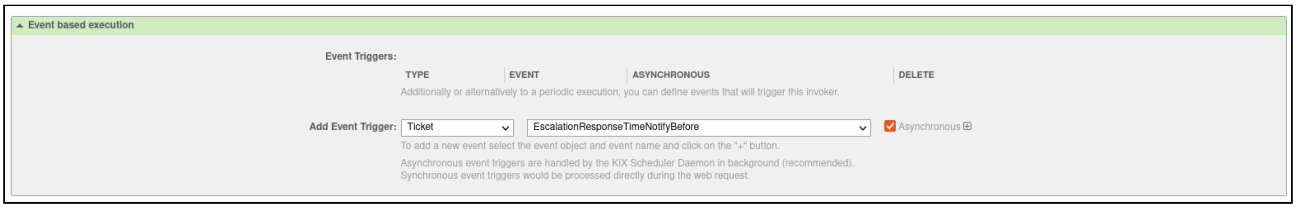
Figure: Interaction between Mappings and Data

Automatic execution

Enter minutes, hours, or days here, thereby controlling the point in time when the invoker is to be executed.

Event-based execution

Here select the event that is to trigger the invoker.



Note

Click the PLUS to the right of the "Add event trigger" fields to select a trigger. Then it appears at the top of the list.
If you do not see an event selection, it is due to changes that have not yet been saved. Apply this with "Save".

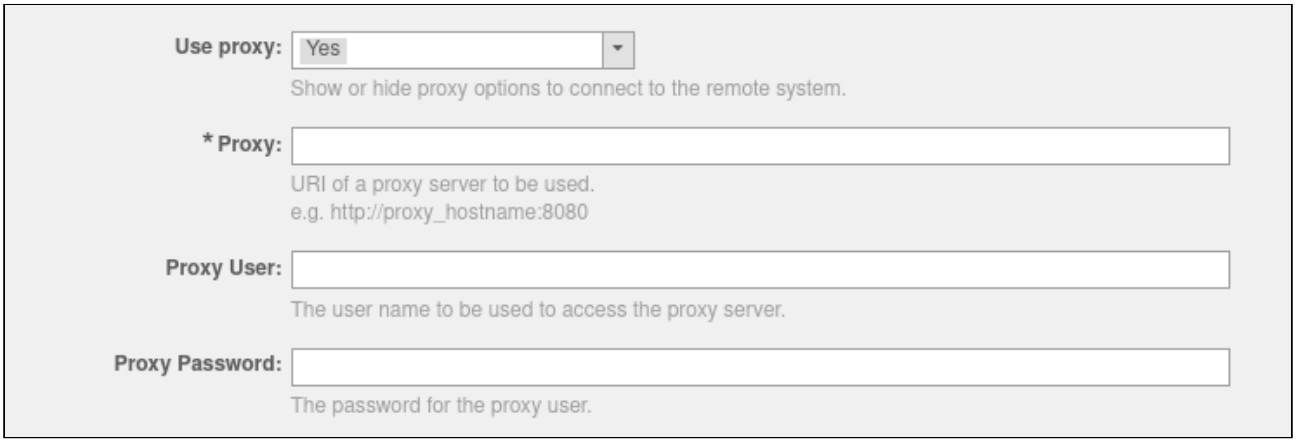
9.3 Expanded Transport Options

9.3.1 Additional Headers

In administration for transport for 'REST' and 'SOAP', additional HTTP headers can be specified. This is not available for LWP as the header for LWP must be defined in the data.

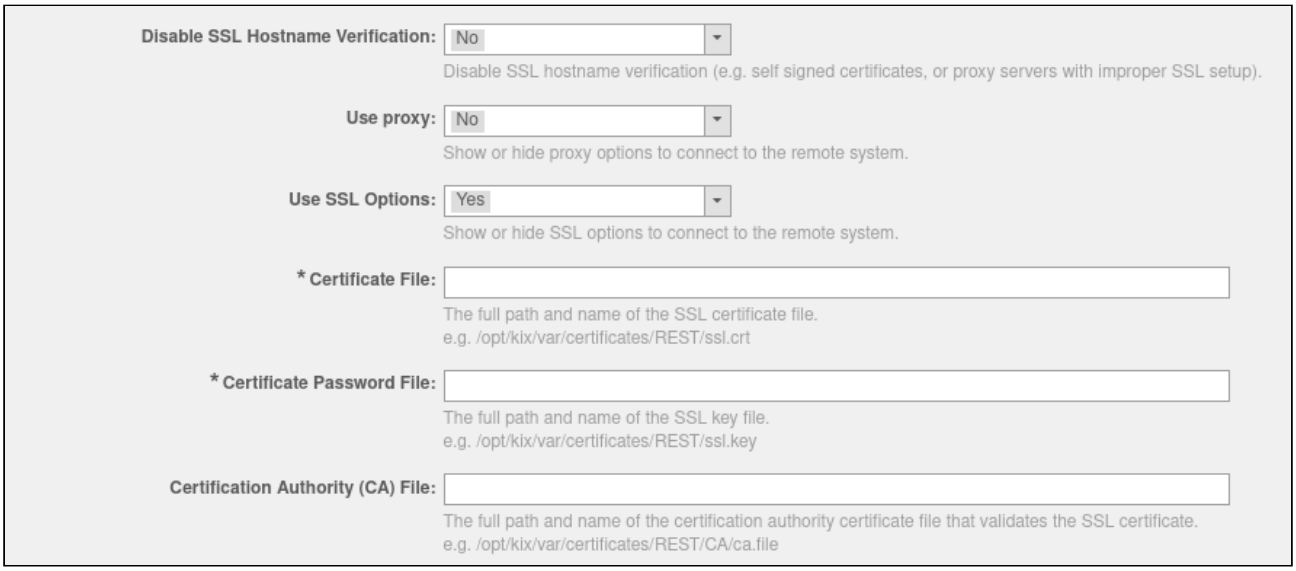
9.3.2 Proxy Settings

In administration for transport for 'LWP', 'REST', and 'SOAP', the use of a proxy can be specified. For 'REST' and 'SOAP', a proxy authentication can be specified. This is not available for LWP as this authentication is sent in the header, and the header for LWP must be defined in the data. The configured proxy is used for HTTP and HTTPS. A configuration 'NO_PROXY' is NOT implemented.



9.3.3 Configuration SSL

In the administration for the transport for 'LWP', 'REST' and 'SOAP', the hostname check for SSL can be deactivated and other SSL configurations can be made.


Disable SSL Hostname Verification:
No
Disable SSL hostname verification (e.g. self signed certificates, or proxy servers with improper SSL setup).
Use proxy:
No
Show or hide proxy options to connect to the remote system.
Use SSL Options:
Yes
Show or hide SSL options to connect to the remote system.
* Certificate File:
The full path and name of the SSL certificate file.
e.g. /opt/kix/var/certificates/REST/ssl.crt
* Certificate Password File:
The full path and name of the SSL key file.
e.g. /opt/kix/var/certificates/REST/ssl.key
Certification Authority (CA) File:
The full path and name of the certification authority certificate file that validates the SSL certificate.
e.g. /opt/kix/var/certificates/REST/CA/ca.file

9.4 Dealing with Large Data Volumes

KIXConnect provides certain optimization mechanisms in order to be able to work more quickly and efficiently when synchronizing data from the interface to KIX.

9.4.1 Caching Processed Datasets

Using a cache, the system should be able to assess whether the processing of data is necessary for a configItem entry, or whether it can be skipped. This applies when using the invoker 'Generic::MethodCall' with the methods from 'Kernel::System::KIXConnect::ITSMConfigItem'.

The following should be noted in the process:

- The element to be used as CacheKey is saved in the mapping.
- A CacheKey is valid globally in the entire web service.
- To have a cache entry created, a TTL (validity period) must be defined. If no TTL is defined, no cache entry is created.
- No TTL has to be defined to check a cache entry.
- If a cache entry is present, data processing including PreCall and PostCall is skipped.
- The cache can be emptied using the command kix.Console.pl Maint::Cache::Delete

Mapping using CacheKeys can look like the following:

```
# Part of ConfigItems-Entry
ConfigItems => {
  Cache => {
    Key => 'Test123', # Identifier for cache entry, valid for every invoker of
current webservice.
    TTL => '14400'    # Time the cache entry is valid in seconds. 14400 seconds => 4
 hours.
                      # If TTL is not given, check for cache entry is performed, but
no cache entry will be created.
                      # Cache entry is created after sucessful processing of config
item entry
  },
  ConfigItem => {
    ...               # Known structure for this part
  }
  ...
}

# or as array
ConfigItems => [
  {
```

```
    Cache => {
      Key => 'Test123', # Identifier for cache entry, valid for every invoker of
current webservice.
      TTL => '14400'    # Time the cache entry is valid in seconds. 14400 seconds =>
4 hours.
                        # If TTL is not given, check for cache entry is performed,
but no cache entry will be created.
                        # Cache entry is created after sucessful processing of config
item entry
    },
    ConfigItem => {
      ...               # Known structure for this part
    }
    ...
  },
  ...
]
```

9.4.2 Delaying/Omitting IncidentStateRecalc

With respect to Config Items, IncidentStateRecalc is a function in KIX which is used to recalculate the incident state when updating an object, based on the LinkTypes and their direction.

During the data processing of 'Kernel::System::KIXConnect::ITSMConfigItem' as module for invoker 'Generic::MethodCall', 'InciStateRecalc' is not to be executed to avoid unnecessarily slowing down the system due to constantly changing data. Once processing is complete, an 'InciStateRecalc' can be executed or completely ignored for all updated ConfigItems.

To control this process, adapt your mappings as follows:

```
# Delay recalc of processed config items till all entries are done
{
  DelayInciStateRecalc => '1',
  ConfigItems => [...],          # or ConfigItems => {...} for only one entry
}

# Skip recalc for every processed config item
{
  SkipInciStateRecalc => '1',
  ConfigItems => [...],          # or ConfigItems => {...} for only one entry
}

# To add processed config items of a PreCall or PostCall to the relevant list for
 DelayInciStateRecalc
{
  DelayInciStateRecalc => '1',
  ConfigItems => [          # or ConfigItems => {...} for only one entry
    {
      PreCall => {
```

```
        ...
        ExtendConfigItemList => 1
      },
      ConfigItem => {...},
      PostCall => {
        ...
        ExtendConfigItemList => 1
      }
    },
    ...
  ]
}
```

9.4.3 Asynchronous Execution of Calls

During data processing of 'Kernel::System::KIXConnect::ITSMConfigItem' as module for invoker 'Generic::MethodCall', for 'PreCall' and 'PostCall' it should be possible to execute the calls asynchronously as independent tasks.

- In the processed entry, `{PreCall}->{Asynchronous}` defines with a true value (1, 2...etc) that the call is executed as an independent task by the scheduler
  - Parameters which evaluate the response (SkipOnFailure and ExtendConfigItemList) are not processed
- In the processed entry, `{PostCall}->{Asynchronous}` defines with a true value (1, 2...etc) that the call is executed as an independent task by the scheduler
  - Parameters which evaluate the response (ExtendConfigItemList) are not processed

A mapping can be adapted as follows:

```
# Run calls asynchronous
{
  ConfigItems => [          # or ConfigItems => {...} for only one entry
    {
      PreCall => {
        ...
        Asynchronous => 1,
      },
      ConfigItem => {...},
      PostCall => {
        ...
        Asynchronous => 1,
      }
    },
    ...
  ]
}
```

9.5 Pre-Configured Web Services

KIX Connect comes with certain web services already pre-configured. These can be activated as required. In the process, they should be adapted to the applicable circumstances, however.

9.5.1 KIX2KIX

With "KIX2KIX" you have the option of connecting two KIX instances with one another. One KIX system is a requester here and the other KIX system is a provider. It is therefore possible to perform the following actions from KIX in another KIX system:

- Create tickets
- On the ticket that has been created:
  - To create articles
  - To change the priority
  - To add attachments (when creating an article)

To do so, the "KIXConnect" package has to be installed on both systems. The package provides the configuration of the web services for KIX as a requester and KIX as a provider.
The package also provides a new sidebar widget "External ticket information" in the ticket zoom view.

In this widget, information about the external ticket is displayed.

Note

The widget is still inactive after installation. If you would like to activate it, place a in the SysConfig key "Frontend::KIXSidebarBackend###KIXSBGIItem002".

9.5.1.1 Prerequisites

- In the "leading" KIX system, the web service "KIX2KIXRequester" has to be activated.
- In the "led" KIX system, the web service "KIX2KIXProvider" has to be activated.
- The queue designation has to be identical in both systems.
- The login name for the contact person has to be identical in both systems.

9.5.1.2 Activating "KIX2KIXRequester" Web Service on Requester System

- In main menu, click "Admin" entry.
- In "System Administration" area, click "Web Services" option.

Web Service Management - Overview opens.

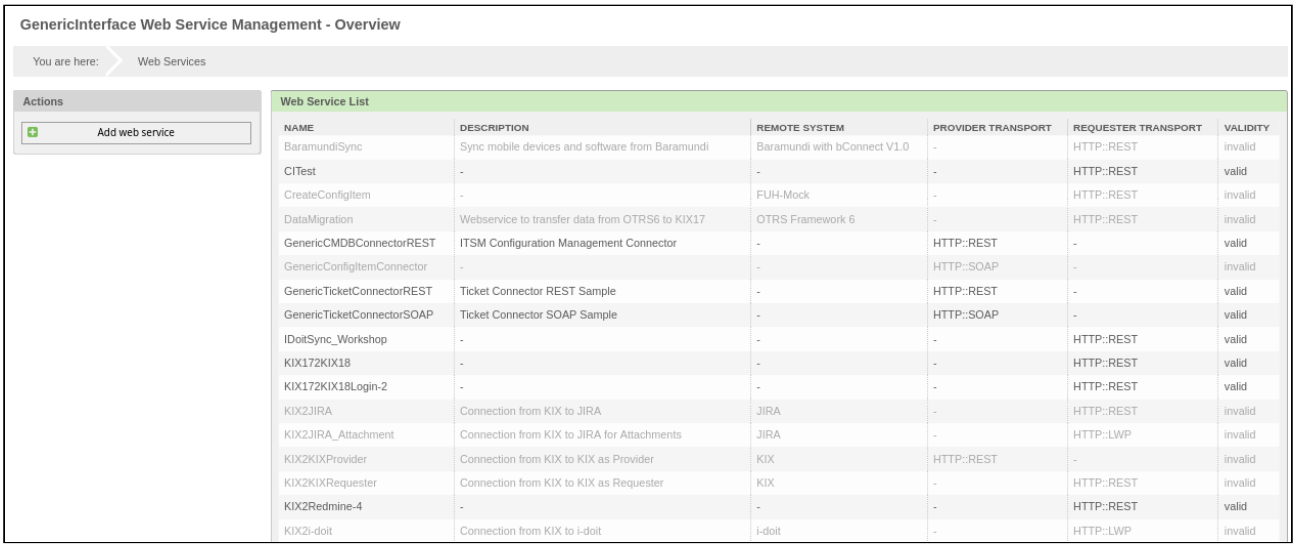
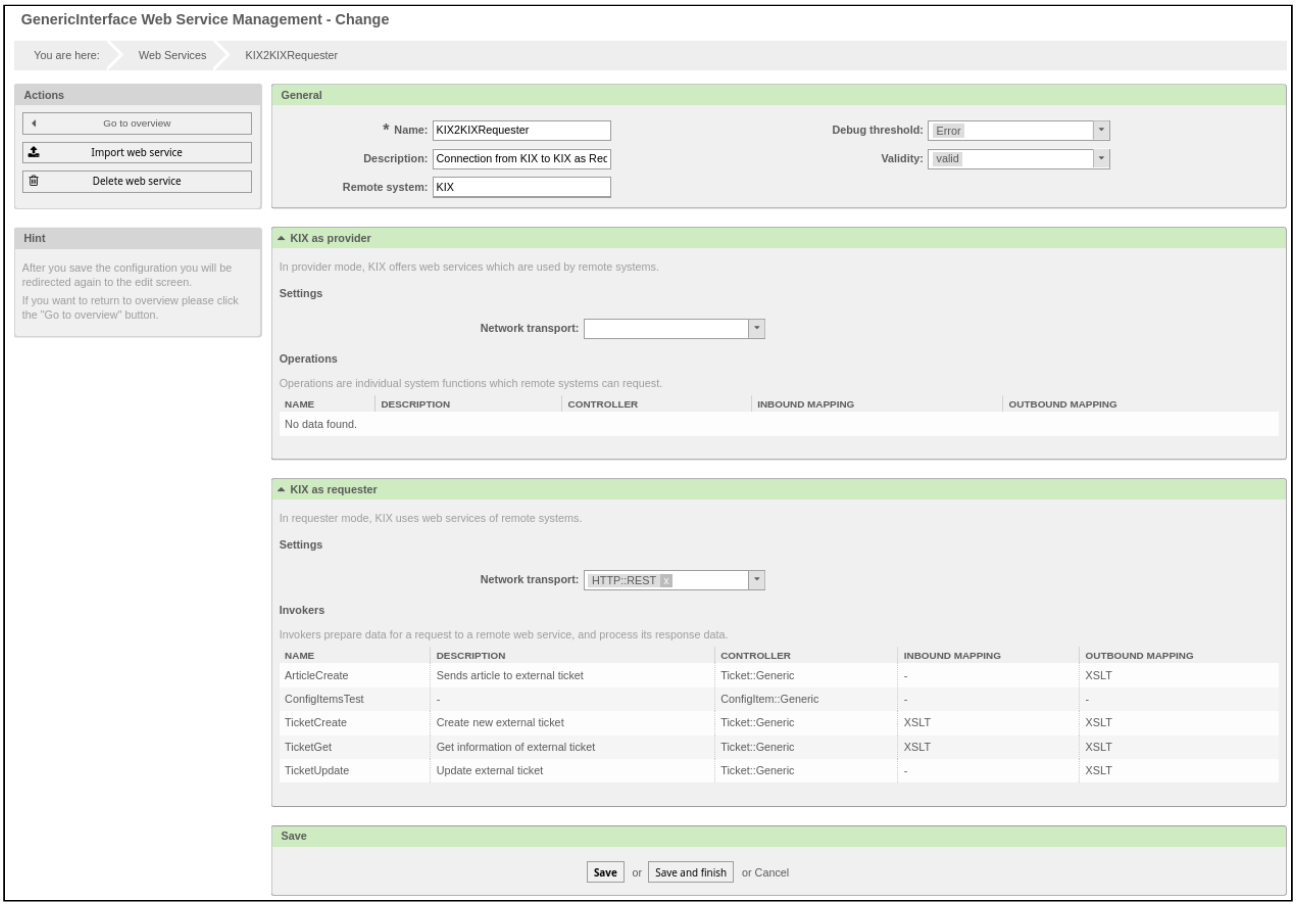


Fig.: Web Service Management - Overview

- Click entry "KIX2KIXRequester" in the list.

"Web Service Management - Change" area opens.



- In the drop-down menu "Validity", select value "valid".
- Finally, click "Save" button.

- In "KIX as requester" widget, click "Configure" button.

"Network transport" widget opens.

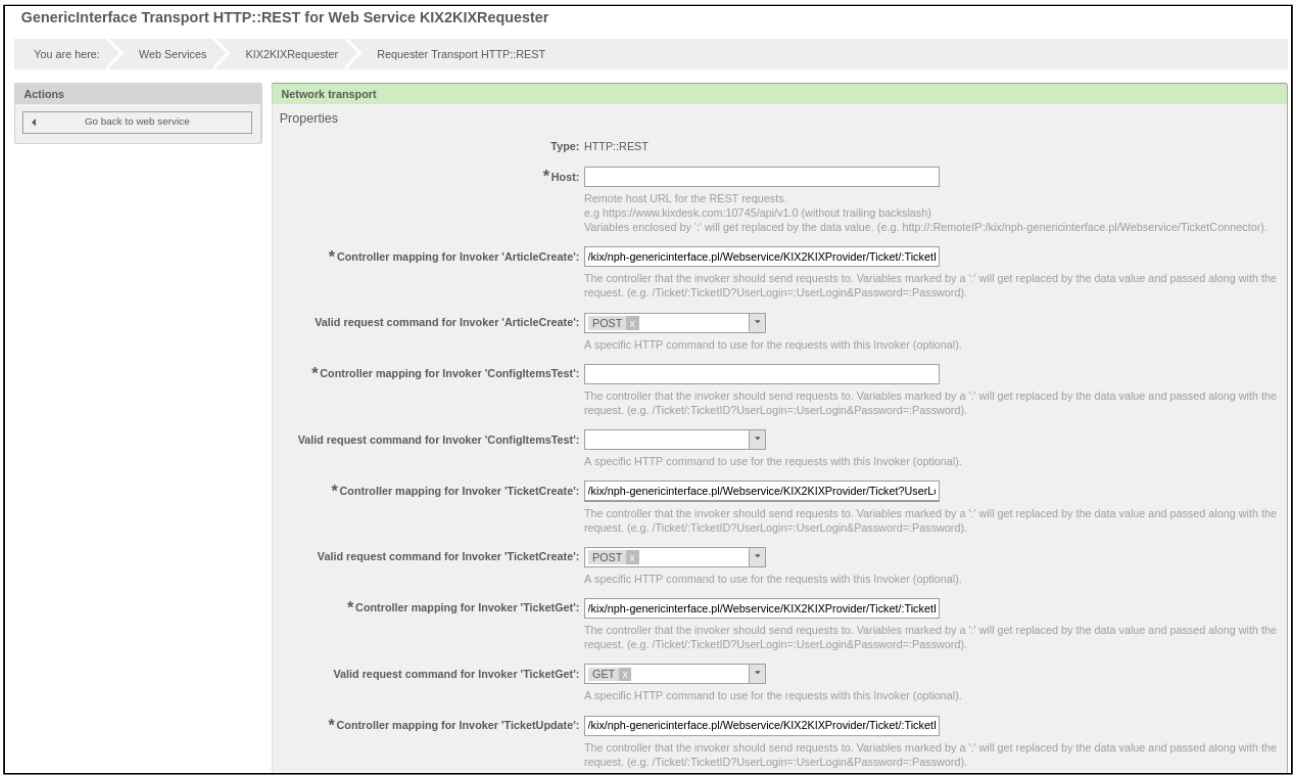


Fig.: "Network transport" Widget

- In "Host" field, enter the URL of the remote host (URL of provider KIX).
- In the fields:
  - Controller mapping for Invoker 'ArticleCreate'
  - Controller mapping for Invoker 'TicketCreate'
  - Controller mapping for Invoker 'TicketGet'
  - Controller mapping for Invoker 'TicketUpdate'

  enter the login name and the password for the provider KIX for <USERNAME> and <PASSWORT (PASSWORD)>.
- Finally, click "Save" button.

9.5.1.3 Activating "KIX2KIXProvider" Web Service on Provider System

- In main menu, click "Admin" entry.
- In "System Administration" area, click "Web Services" option.

Web Service Management - Overview opens.

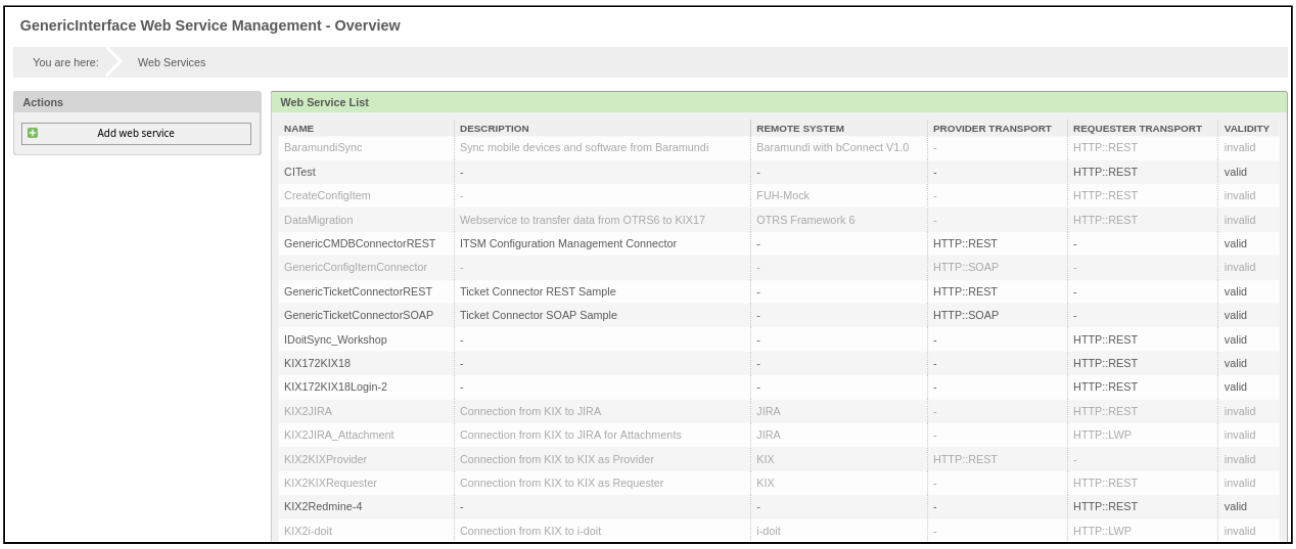


Fig.: Web Service Management - Overview

- Click entry "KIX2KIXProvider" in the list. "Web Service Management - Change" area opens.

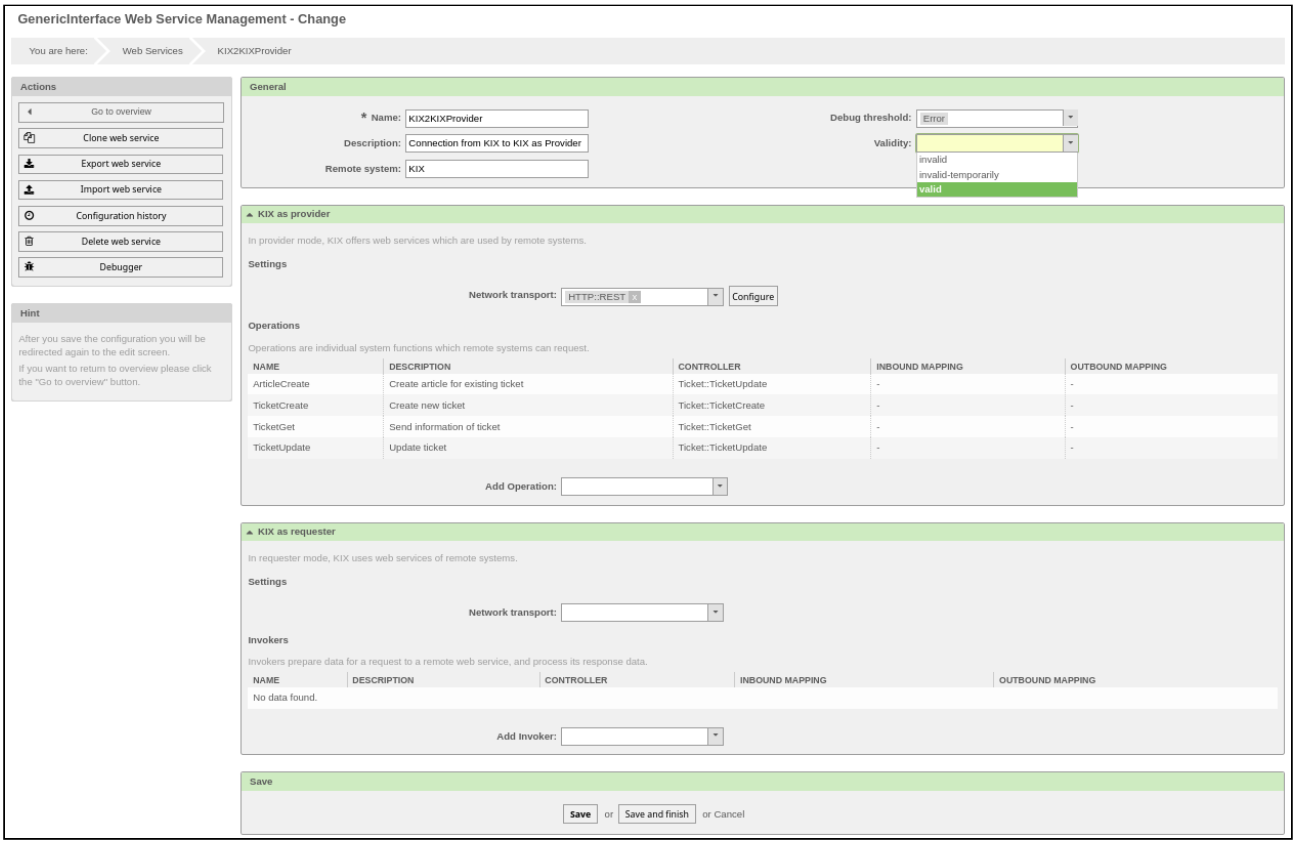


Fig.: Web Service Management - Change

- In the drop-down menu "Validity", select value "valid".
- Finally, click "Save" button.

> **Note**
>
> The maximum message length is defined as 24000000 bytes by default. This setting is especially relevant for the transmission of attachments. Requests that are larger than this setting are rejected with a message in the provider's debugger. If you want to change the default value, click "Configure" button in "KIX as provider" widget. "Network transport" widget then opens. In "Maximum message length", specify the desired value and click "Save".

9.5.1.4 Creating a New Ticket from Requester KIX in Provider KIX

- In "Tickets" menu, select "New ticket" entry.
- Fill in the input fields as when creating a ticket.
- Place a in "Create external ticket" field.
- Finally, click button "Create".

The ticket has now been created with the specified data by requester KIX in provider KIX.

In the standard version of the package, the following ticket/article attributes are transmitted during ticket creation:

Ticket	Article	Attachment
Title	ArticleType	Filename
Type	Sendertype	ContentType
Queue	From	Content (Base64)
State	To	Disposition
Priority	Subject	ContentID
CustomerUserID	Body	
CustomerID	ContentType	
	HistoryType	

The attributes to be set during ticket creation have to have the same designation in both systems. If they do not have the same designations, there will be an error message in the provider system's debugger.

9.5.1.5 Creating a New Article in an External Ticket

If you want to add an article to an external ticket, you have two options:

- Using "Edit" button or
- Using "Note" tab in the ticket zoom view

If you click "Edit" button, the dialog "Add Note to Ticket#.... " opens.

Here you have the option of drafting the article via "Subject"/"Text".

Fill out the fields in the dialog and place a by "Send to external ticket". Finally, click button "Submit".
In this dialog, you can also change the state and priority of the external ticket. However, this option is subject to the limitation that a new article is always generated via this dialog. If you want to change the priority or the state without generating a new article, then the method described below under "Changing the Priority in an External Ticket" is recommended.
If you use the note tab, you only have the option of creating an article in an external ticket by filling out the fields "Subject"/"Text". There is not the option of changing the state and priority here.

9.5.1.6 Changing the Priority in an External Ticket

You can change the priority in the external ticket by using the "Ticket Core Data" tab in the requester KIX.

Change the priority here and, finally, click button "Submit". The priority is changed accordingly in the external ticket.
The changed value of the external ticket can be seen in the sidebar widget "External ticket information".

9.5.2 KIX2JIRA

With "KIX2JIRA" you have the option of connecting KIX with JIRA. Here, KIX as the requester functions as the active/leading system, whilst JIRA functions as the provider. It is therefore possible to perform the following actions in JIRA from KIX:

- To generate a new process (issue) in JIRA if a ticket has been created in KIX.
- To add a comment to a process in JIRA.
- To update an existing JIRA process.
- To transmit an attachment to JIRA.

Consists of the web services:

- KIX2JIRA
- KIX2JIRA_Attachment

Please note

If attachments are to be transmitted, attachments must be permitted for processes for the relevant project in JIRA.

Note

In the event of specific changes that deviate from the standard configuration, this may result in an error message from the system. We recommend you contact our support team to rectify the error.

9.5.2.1 Activating "KIX2JIRA" Web Service

- In main menu, click "Admin" entry.
- In "System Administration" area, click "Web Services" option.

Web Service Management - Overview opens.

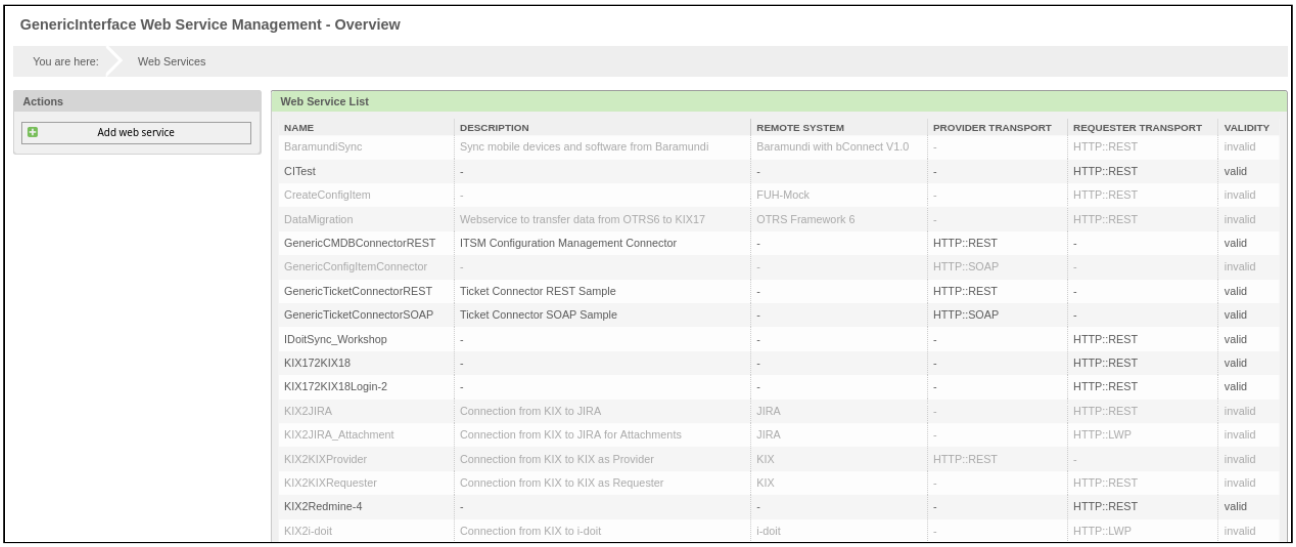


Fig.: Web Service Management - Overview

- Click entry "KIX2JIRA" in the list. The "Web Service Management - Change" area opens

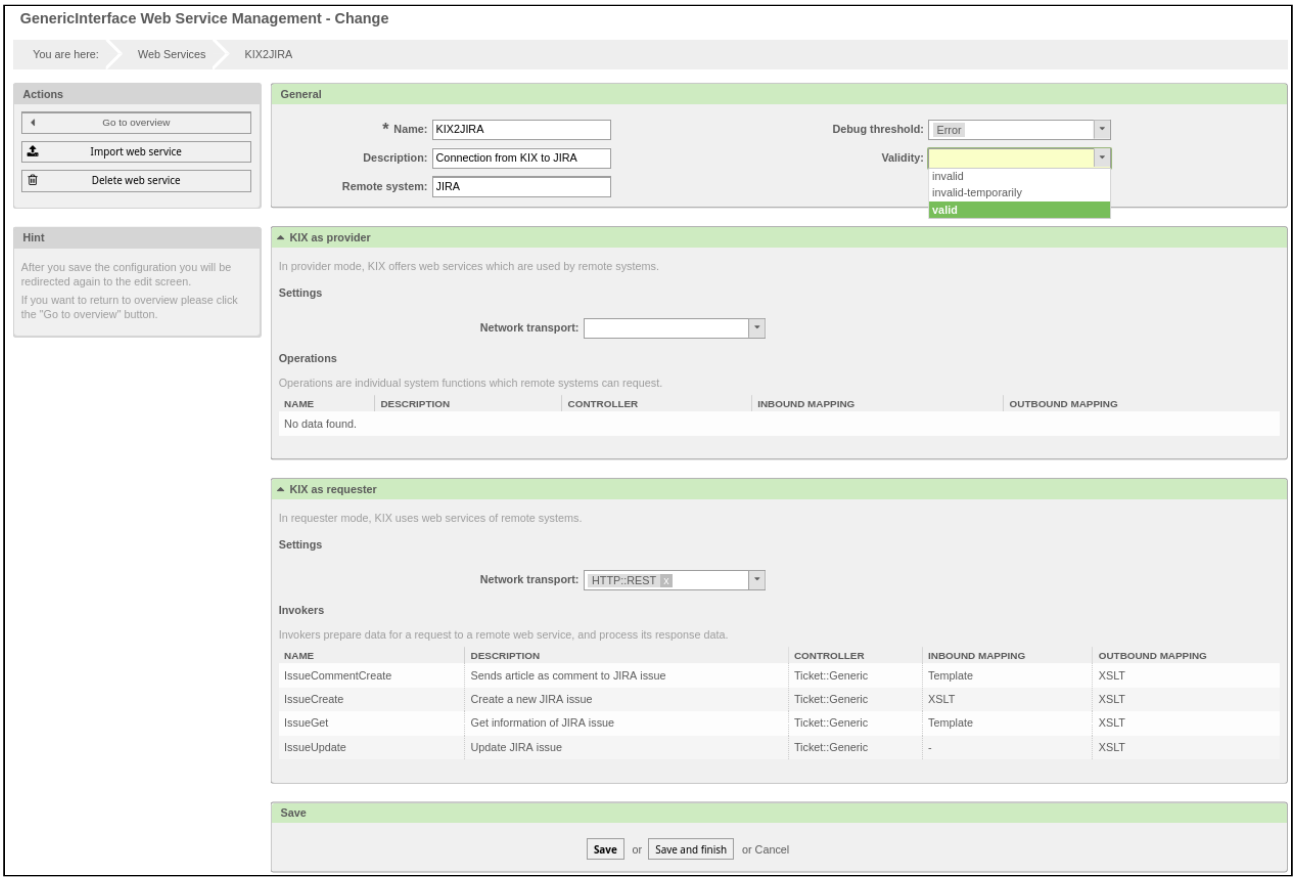


Fig.: Web Service Management - Change

- In the drop-down menu "Validity", select value "valid".

- Finally, click "Save" button.
- In "KIX as requester" widget, click "Configure" button.

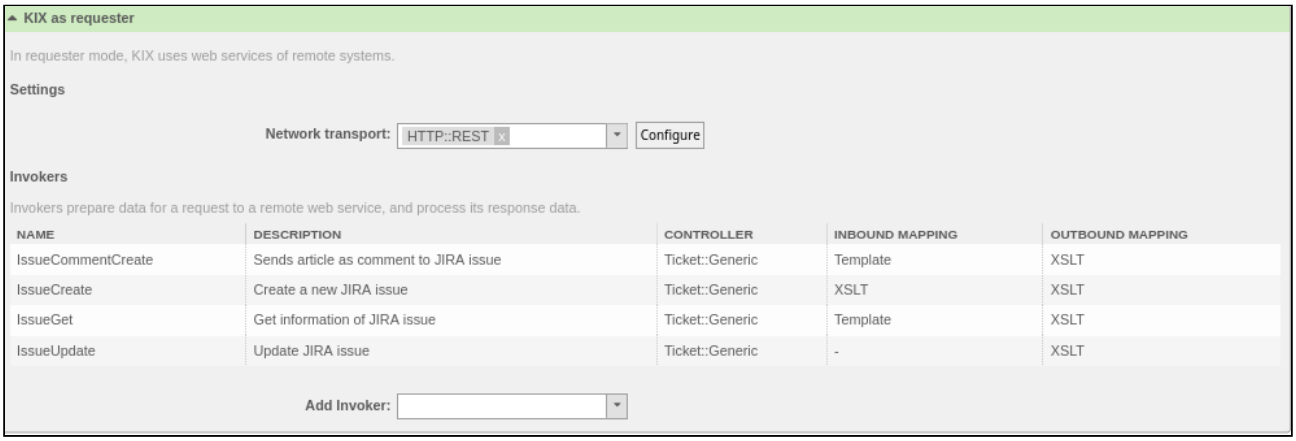


Fig.: "KIX as requester" Widget

"Network transport" widget opens.

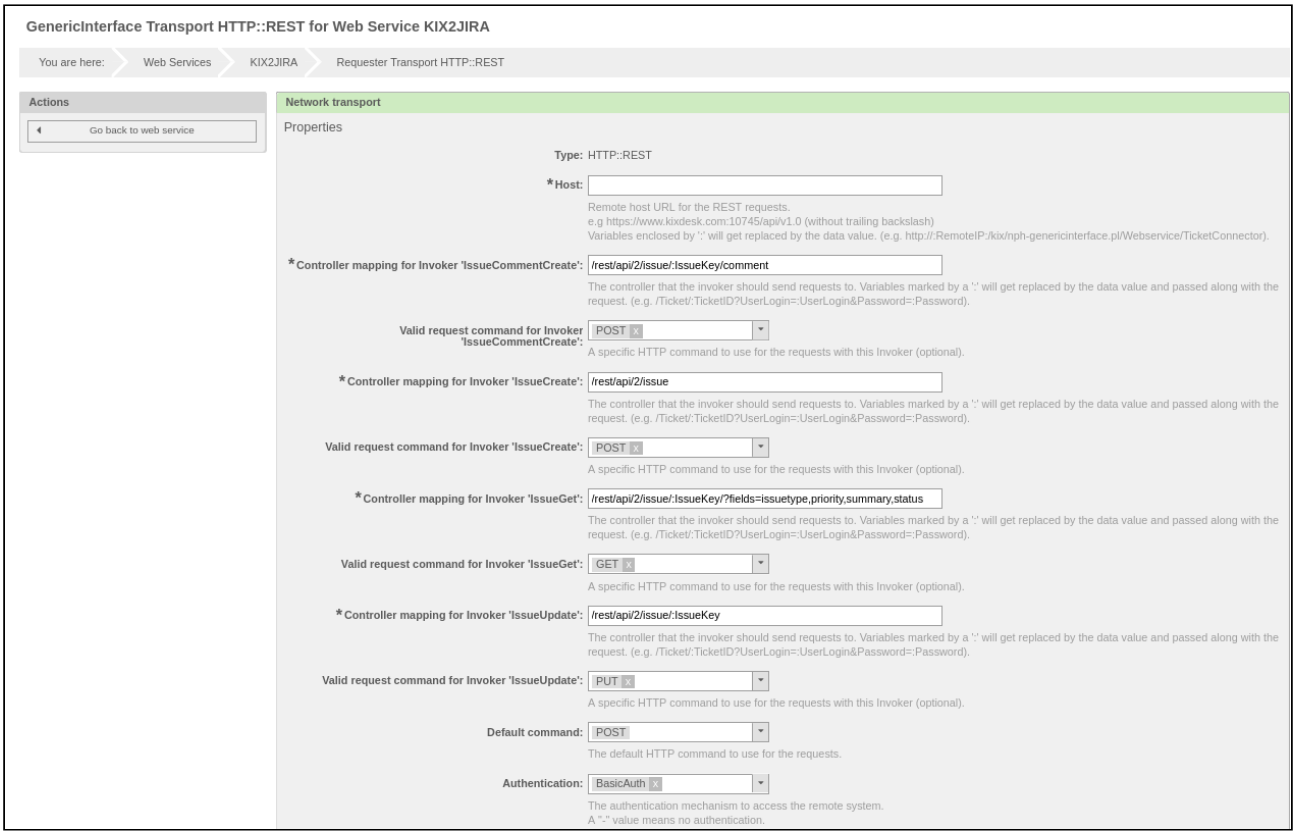


Fig.: "Network transport" Widget

- In "Host" field, specify URL of remote host (in this case JIRA).

- Specify the user name with which you want to log into the remote host in "User" field. (In this case the JIRA user name).
- Specify the password with which you want to log into the remote host in "Password" field. (In this case the JIRA password).
  All other settings have already been made through "KIX Connect" module and do not have to be changed.
- Finally, click "Save" button.

9.5.2.2 Activating "KIX2JIRA_Attachment" Web Service

- In main menu, click "Admin" entry.
- In "System Administration" area, click "Web Services" option.

Web Service Management - Overview opens.

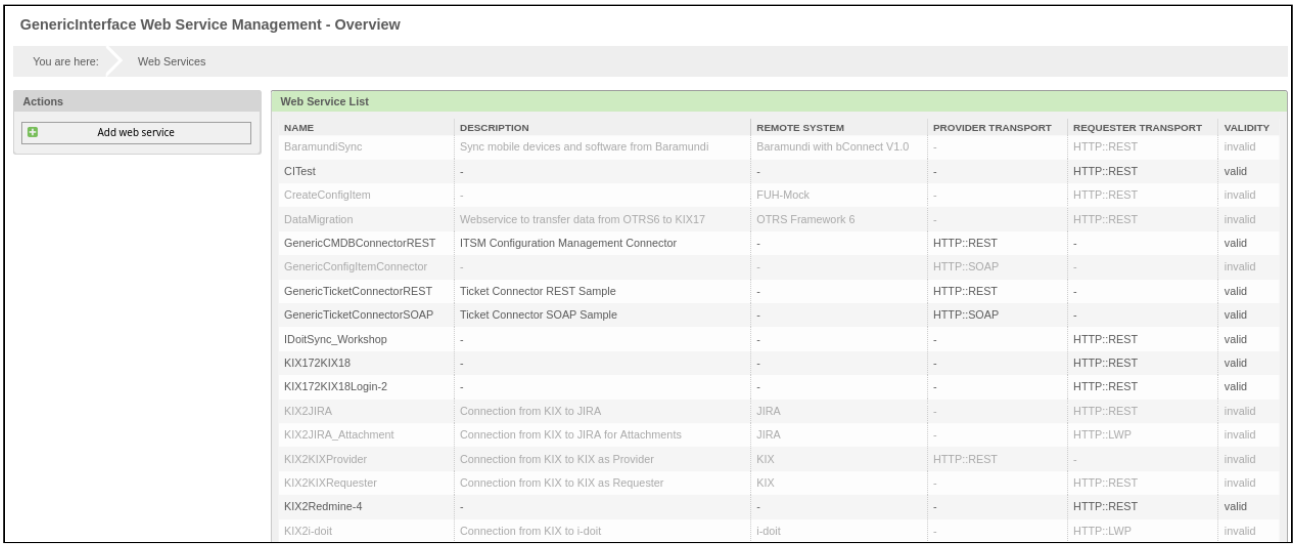


Fig.: Web Service Management - Overview

- Click entry "KIX2JIRA" in the list. "Web Service Management - Change" area opens.

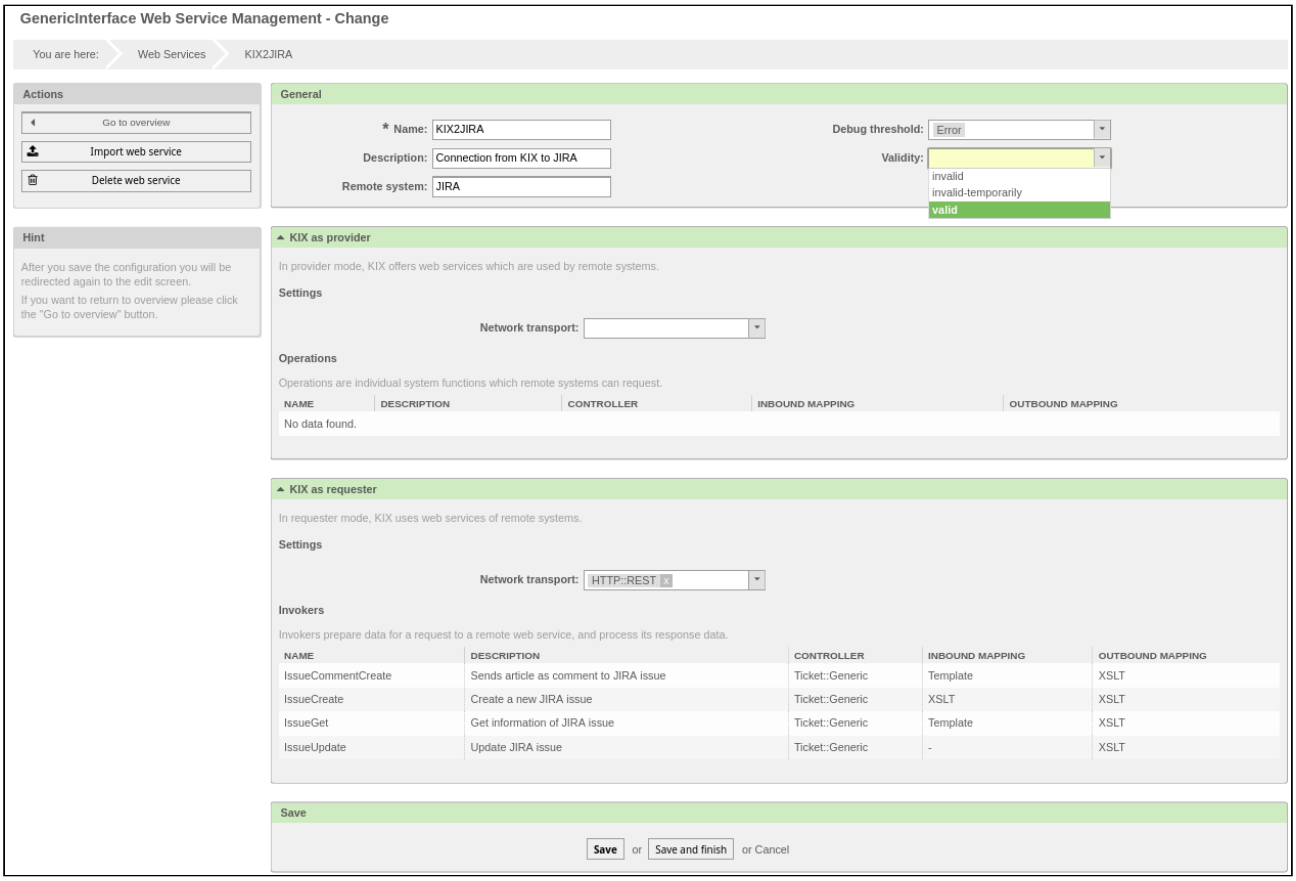


Fig.: Web Service Management - Change

- In the drop-down menu "Validity", select value "valid".
- Finally, click "Save" button.
- In "KIX as requester" widget, click "Configure" button.

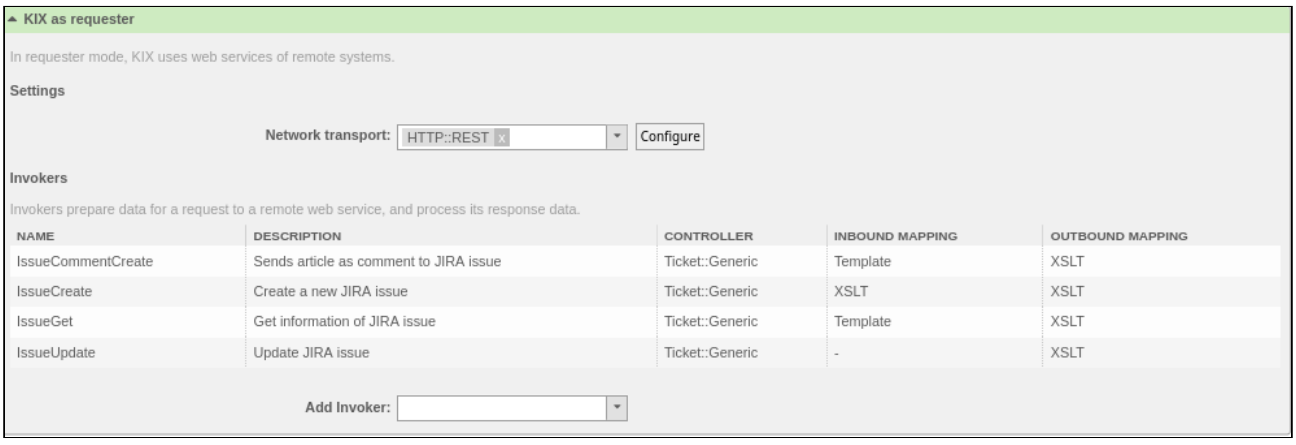


Fig.: "KIX as requester" Widget

"Network transport" widget opens.

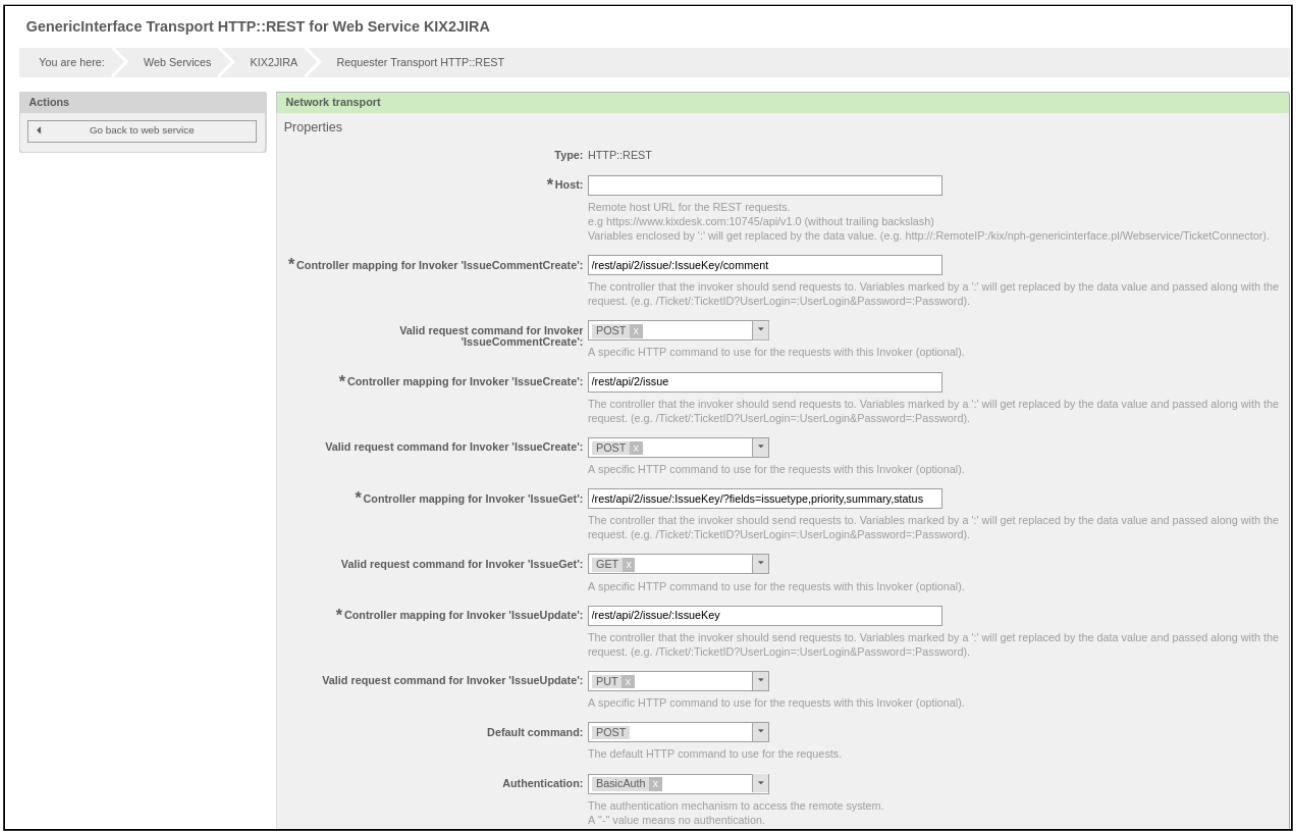


Fig.: "Network transport" Widget

- In "Host" field, specify URL of remote host (in this case JIRA).
- Specify the user name with which you want to log into the remote host in "User" field. (In this case the JIRA user name).
- Specify the password with which you want to log into the remote host in "Password" field. (In this case the JIRA password).
  All other settings have already been made through "KIX Connect" module and do not have to be changed.
- Finally, click "Save" button.

9.5.2.3 "JIRA information" Sidebar Widget

By installing the "KIX Connect" module, a new sidebar widget "JIRA information" has been created for the ticket zoom view. The widget shows the following information:

- "Title" = the name of the process in JIRA
- "Key" = designation of the JIRA project key
- "Type" = type of JIRA process (JIRA issue)

- "State" = state of the JIRA process
- "Priority" = priority of the JIRA process

In the delivery state, the widget is still set to inactive. Carry out the following steps if you want to activate it:

- In main menu, click "Admin" entry.
- In "System Administration" area, click "SysConfig" entry.
- In the drop-down menu, click "KIXConnect" group.
- Select subgroup "Frontend::Agent::KIXSidebarGIView".
- Set a  by the entry "Frontend::KIXSidebarBackend###KIXSBGIItem001" and click "Update" button.

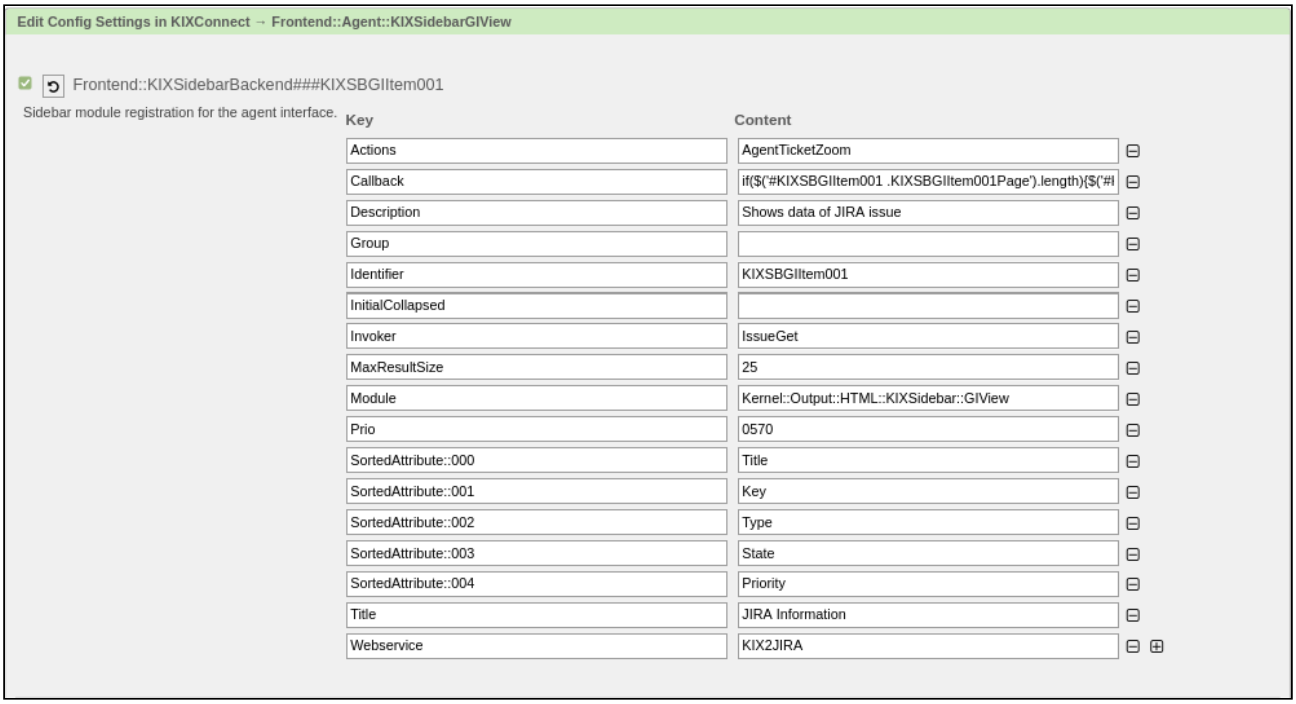


Fig.: SysConfig – Key "Frontend::KIXSidebarBackend###KIXBGIItem001"

9.5.2.4 Creating a Ticket in KIX and Transmitting Data to JIRA

- In "Tickets" menu, select "New ticket" entry.
- Fill in the input fields as when creating a ticket.
- In the drop-down menu "Type of JIRA issue", select a value from the list.
- In "Project of JIRA issue" field, enter project key of JIRA project.
- Finally, click button "Create".

The ticket has now been created in KIX with the specified data and a process has been created in JIRA according to the specified data. When you retrieve the ticket zoom view of the ticket that has just been created (for example by clicking the ticket number in the green system message), you can see in the "JIRA information" widget that the data has been successfully transmitted to JIRA. You can also open the process in JIRA to check.

> **Note**
>
> If the new ticket was created in the KIX, the button "Send to JIRA" can still be seen on the first article of the ticket, although the data has already been transferred to JIRA.
> Background: The text content of the KIX ticket, and thus of the first article, is identical to the description of the process in the JIRA issue. Therefore, no further comment with the same content needs to be created in Jira.

9.5.2.5 Changing a JIRA Process Title

- Call up the ticket zoom view for the ticket in KIX.
- Click the "Ticket Core Data" tab.
- Specify the changed designation for the process title in "Title" field.
- Finally, click button "Submit".

Process title has now been updated in KIX and JIRA. In "JIRA information" widget, you can see the changed title.

9.5.2.6 Transmitting a Note in KIX as a JIRA Comment

- Call up the ticket zoom view for the ticket in KIX.
- Click "Edit" button or "New Note" tab.

If the package has been set up correctly and a reference to JIRA is available, then a checkbox "An JIRA senden (Send to JIRA)" now appears in the editing mode.

- Fill out the fields "Subject" and "Text" and mark "Send to JIRA" with  . Optionally, you have the possibility here to transmit an attachment with the note to JIRA.
- Finally, click button "Submit".

The data is transmitted from KIX to JIRA. The "Subject" is shown in JIRA as the first line in the comment field. After a line break, the content from "Text" field is then shown in the comment field. The attachment is placed in JIRA in the "Attachments" area. When transmitting attachments, you can also select and transmit multiple attachments.

9.5.3 X2i-doit

i-doit (I document IT) is a web-based free software that is intended to document complex IT infrastructures and which is supplemented with additional functions from a configuration management database (CMDB). The elements of the IT infrastructure are created as objects and provided with attributes.

KIX Connect provides integration of the i-doit device database (i-doit CMDB) in ticket processing in KIX. This is carried out by setting up a form field which draws its selection values from the i-doit CMDB. The field can contain multiple entries. The number of permissible entries can be configured.

In the input screen and "Dynamic Fields" ticket tab in the ticket view, you can see a field 'Betroffene CIs (Affected CIs)'. Using autocomplete, you can search for entries from i-doit here and select them. The 'Betroffene CIs (Affected CIs)' saved for the ticket are then displayed accordingly in the 'Betroffene CIs (Affected CIs)' sidebar in the ticket view. In the case of multiple selections being made, the individual inputs are shown as pages in the sidebar. The attributes displayed can be configured.

> **Note**
>
> No ConfigItem information is synchronized in the KIX CMDB.

With "KIX2i-doit", you have the option of providing data from i-doit for ticket processing in KIX.

In doing so, it is possible:

- To note objects from i-doit as 'Betroffene CIs (Affected CIs)' on a ticket
- To display information concerning noted objects on the ticket

Consists of the web services:

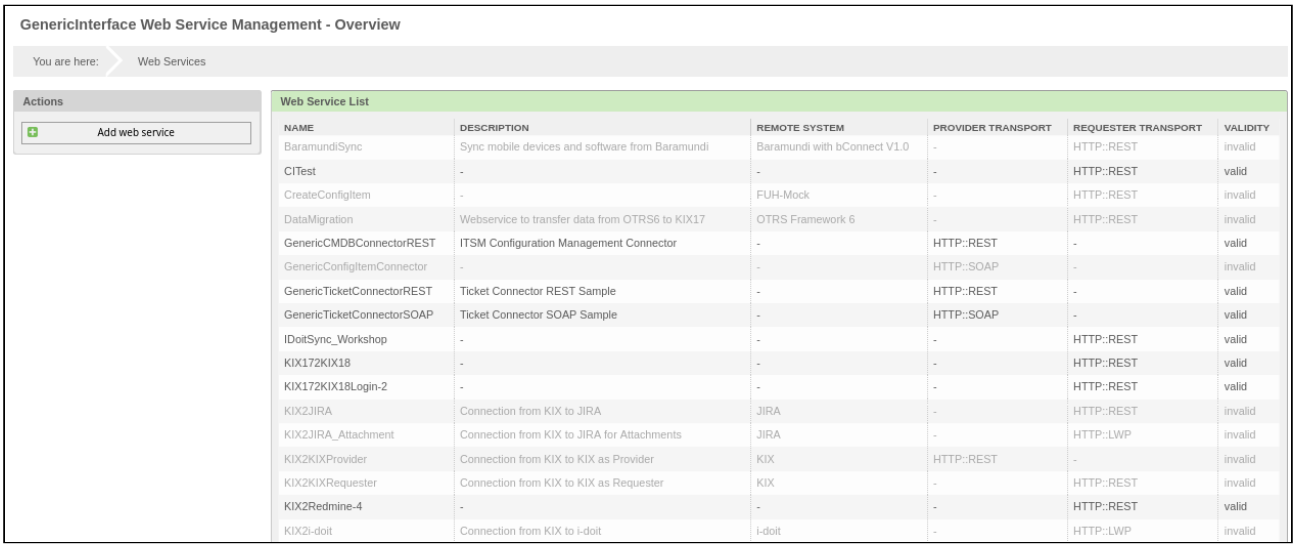
- KIX2i-doit

9.5.3.1 Activating "KIX2i-doit" Web Service

In the delivery state, the widget is still set to inactive. In order to activate it, carry out the following steps:

- In main menu, click "Admin" entry.
- In "System Administration" area, click "SysConfig" entry.
- In the drop-down menu, click "KIXConnect" group.
- Select "KIX2i-doit" subgroup.
- Fill out the configuration items on this page and click "Update".
- Then, in main menu, click "Admin" entry.
- In "System Administration" area, click "Web Services" option.

Web Service Management - Overview opens.



NAME	DESCRIPTION	REMOTE SYSTEM	PROVIDER TRANSPORT	REQUESTER TRANSPORT	VALIDITY
BaramundiSync	Sync mobile devices and software from Baramundi	Baramundi with bConnect V1.0	-	HTTP::REST	invalid
CITest	-	-	-	HTTP::REST	valid
CreateConfigItem	-	FUH-Mock	-	HTTP::REST	invalid
DataMigration	Webservice to transfer data from OTRS6 to KIX17	OTRS Framework 6	-	HTTP::REST	invalid
GenericCMDBConnectorREST	ITSM Configuration Management Connector	-	HTTP::REST	-	valid
GenericConfigItemConnector	-	-	HTTP::SOAP	-	invalid
GenericTicketConnectorREST	Ticket Connector REST Sample	-	HTTP::REST	-	valid
GenericTicketConnectorSOAP	Ticket Connector SOAP Sample	-	HTTP::SOAP	-	valid
IDoitSync_Workshop	-	-	-	HTTP::REST	valid
KIX172KIX18	-	-	-	HTTP::REST	valid
KIX172KIX18Login-2	-	-	-	HTTP::REST	valid
KIX2JIRA	Connection from KIX to JIRA	JIRA	-	HTTP::REST	invalid
KIX2JIRA_Attachment	Connection from KIX to JIRA for Attachments	JIRA	-	HTTP::LWP	invalid
KIX2KIXProvider	Connection from KIX to KIX as Provider	KIX	HTTP::REST	-	invalid
KIX2KIXRequester	Connection from KIX to KIX as Requester	KIX	-	HTTP::REST	invalid
KIX2Redmine-4	-	-	-	HTTP::REST	valid
KIX2i-doit	Connection from KIX to i-doit	i-doit	-	HTTP::LWP	invalid

Fig.: Web Service Management - Overview

- Click entry "KIX2i-doit" in the list. The "Web Service Management - Change" area opens

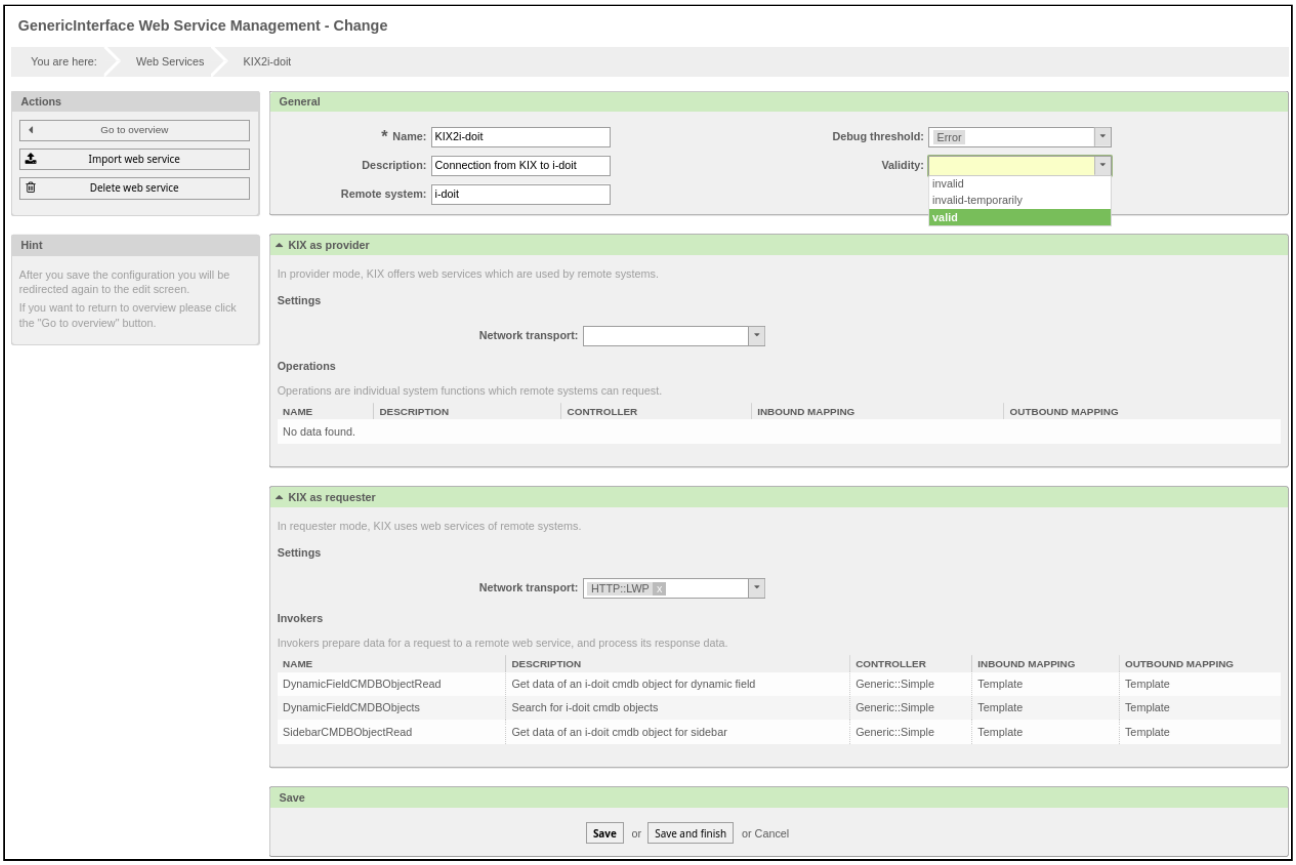


Fig.: Web Service Management - Change

- In the drop-down menu "Validity", select value "valid".
- Finally, click "Save" button.

No changes have to be made in the web service when HTTP is used. If HTTPS is to be used, in the web service 'KIX2i-doit'

- '[http://' must be changed to '[https://' in the transport configuration for 'Host'.
- '[http://' must be changed to '[https://' in the mapping for incoming response data of the invoker 'SidebarCMDBObjectRead' in the row with 'SYSID'.

9.5.3.2 "Affected CIs" Sidebar Widget

By installing the "KIX Connect" module, a new sidebar widget "Affected CIs" has been created for the ticket zoom view. The widget shows the following information:

- General information
  - "Category" – category of the object in i-doit (infrastructure, software, contacts, other)

    - "Type" – type of the object in i-doit (building, server, telephone, monitor, etc.)
    - "SYSID" – the system ID of i-doit, with a link to the object
    - "Title" – name of the object
    - "State" – CMDB state of the object
    - "Use" – what the object is being used as (productively, quality assurance)
    - "Tags" – keywords assigned to the object
    - "Description"
    - "Created" – creation date of the object
    - "Changed" – date on which the object was last changed
- Model information
    - "Manufacturer"
    - "Model"
    - "Product ID"
    - "Serial number"
- Address information
    - "Street"
    - "Number" – house number on the street
    - "ZIP code"
    - "City"
    - "Region"
    - "Country"

In the delivery state, the widget is still set to inactive. Carry out the following steps if you want to activate it:

- In main menu, click "Admin" entry.
- In "System Administration" area, click "SysConfig" entry.
- In the drop-down menu, click "KIXConnect" group.
- Select subgroup "Frontend::Agent::KIXSidebarGIView".
- Place a  by the entry "Frontend::KIXSidebarBackend###KIXSBGIItem003" and click "Update" button.

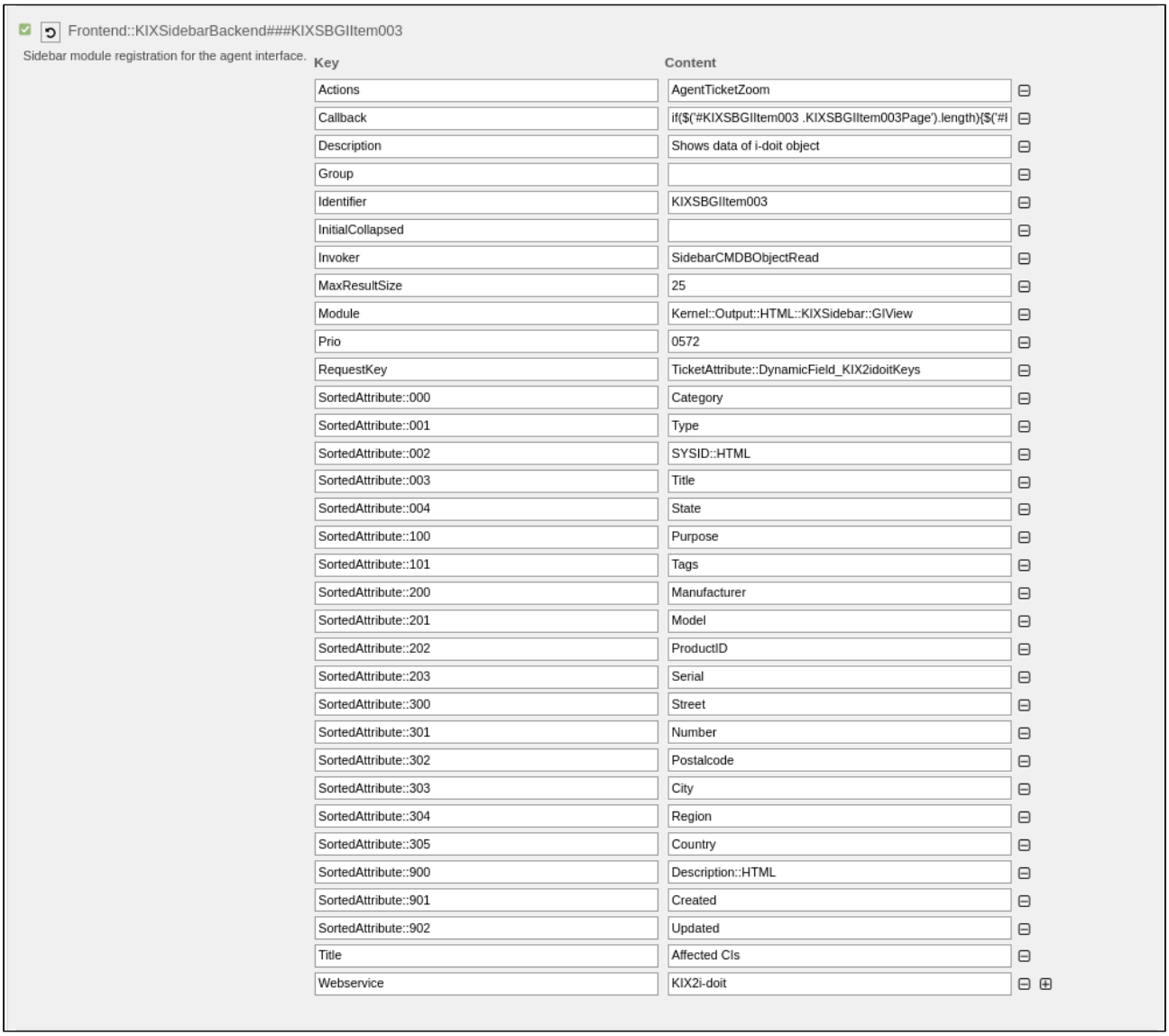


Fig.: SysConfig – Key "Frontend::KIXSidebarBackend###KIXSBGIItem003"

There is also a sidebar for the customer frontend for this connection; this is also inactive upon delivery. Take the following steps to activate it:

- In main menu, click "Admin" entry.
- In "System Administration" area, click "SysConfig" entry.
- In the drop-down menu, click "KIXConnect" group.
- Click subgroup "Frontend::Customer::KIXSidebarGIView".
- Place a  by the entry "CustomerFrontend::KIXSidebarBackend###KIXSBGIItem003" and click "Update" button.

9.5.3.3 Noting Affected CIs from i-doit on the Ticket

- In main menu, click item "New ticket".
- You can find the "Betroffene CIs (Affected CIs)" field when filling out the content. This is an autocomplete field. Enter a search term or part of one here. You can also use * as a wildcard. In the standard version, the field requires at least three characters before a search is started.
- Once the ticket has been created, the affected CI is saved on the ticket.

In order to note an affected CI from i-doit on an existing ticket, open that ticket and click "Dynamic Fields" tab. You can find the field "Betroffene CIs (Affected CIs)" here. Filling this out can be carried out in line with the procedure for creating a new ticket.

9.5.4 BaramundiSync

With BaramundiSync, you can synchronize devices (Device) and their installed software (Software) from Baramundi to the KIX CMDB. For this, a web service template is available that is inactive when delivered. This can request data from the Baramundi interface "bConnect" via REST.

- CI class "Device" is created when KIX Connect is installed.
- CI class "Software" is used by BaramundiSync with the available default attributes.

9.5.4.1 Activating "BaramundiSync" Web Service

In the delivery state, the web service is still set to inactive. In order to activate it, carry out the following points:

- In main menu, click "Admin" entry.
- In "System Administration" area, click "Web Services" option.

The "Web Service Management - Change" area opens

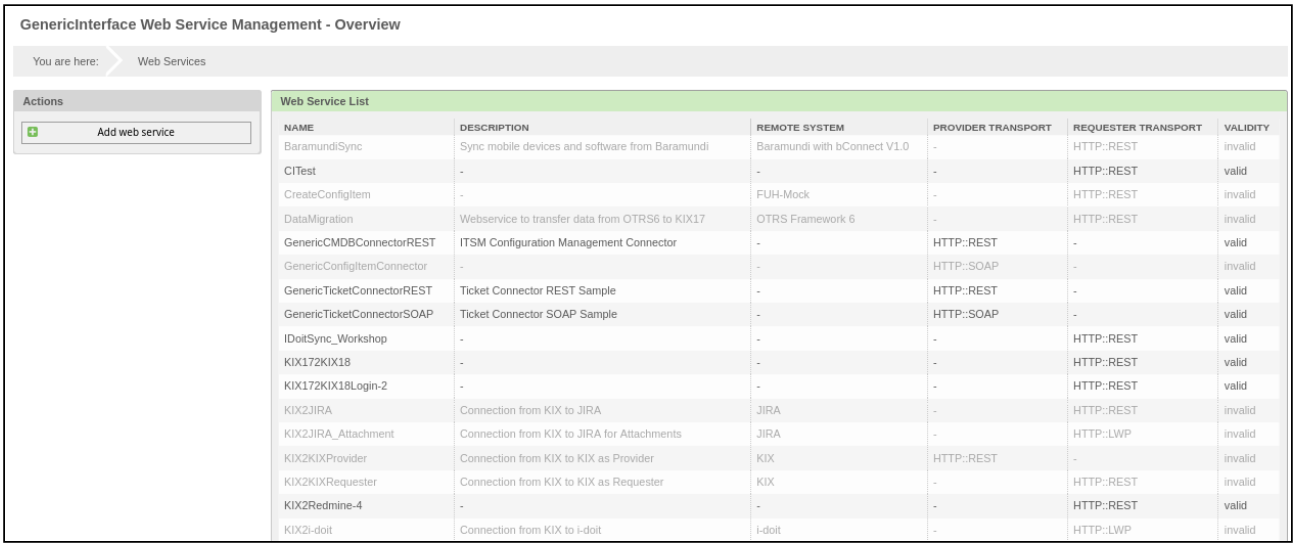


Fig.: Web Service Management - Overview

- Click entry "BaramundiSync" in the list. The "Web Service Management - Change" area opens

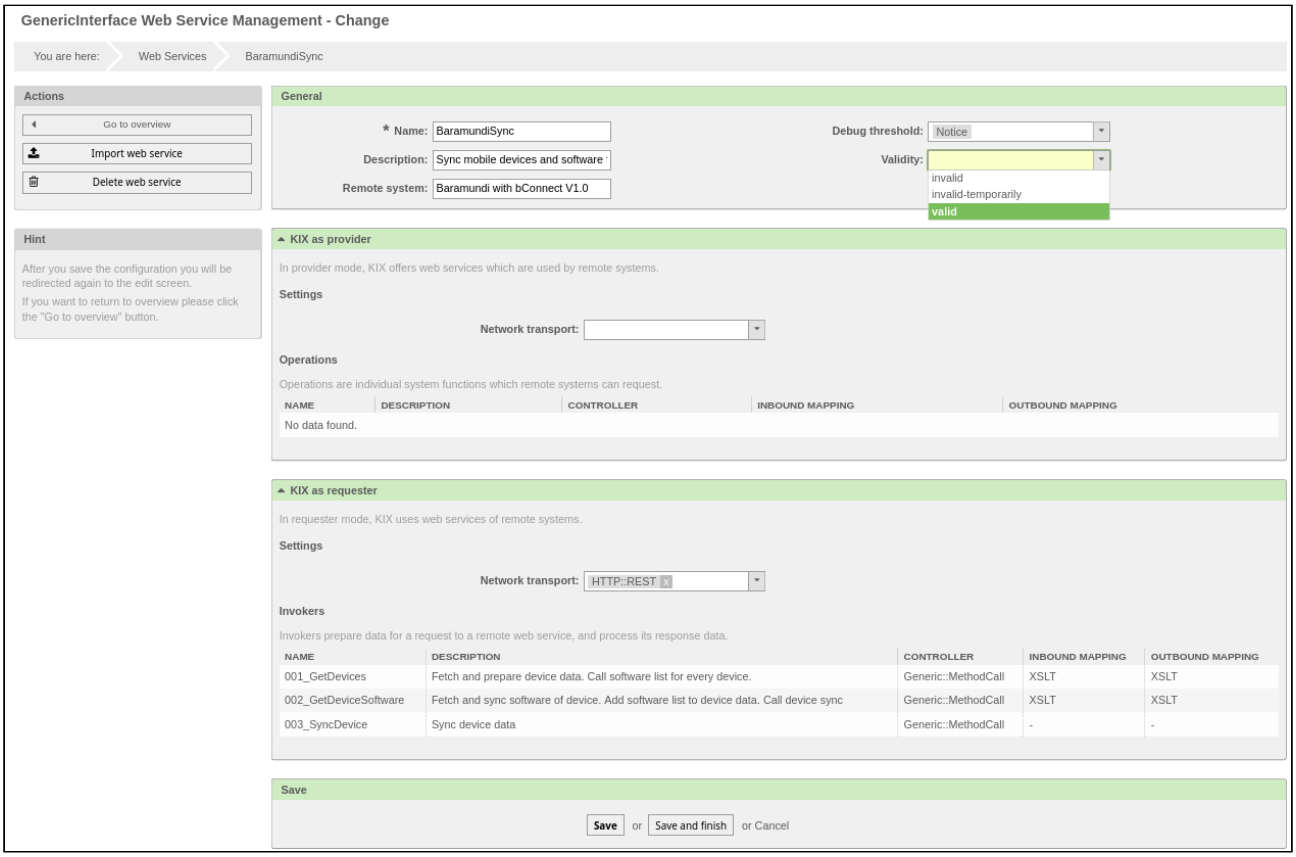


Fig.: Web Service Management - Change

- In the drop-down menu "Validity", select value "valid".
- Finally, click "Save" button.

To adapt "BaramundiSync" to your circumstances, make the applicable modifications to the network transport.

- In "KIX as requester" area, click "Configure".

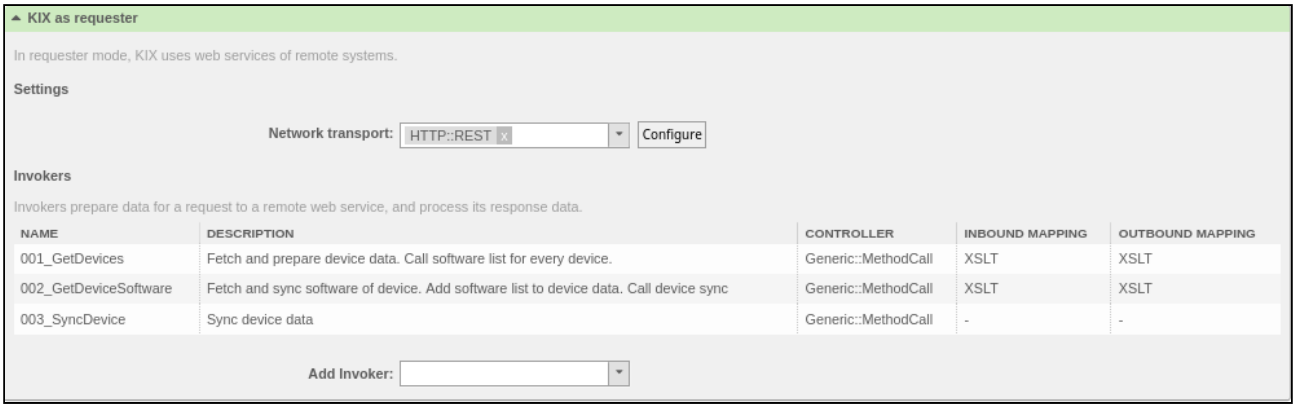


Fig.: Changing Transport Properties

- Fill out the required fields
- Finally, click "Save" button.

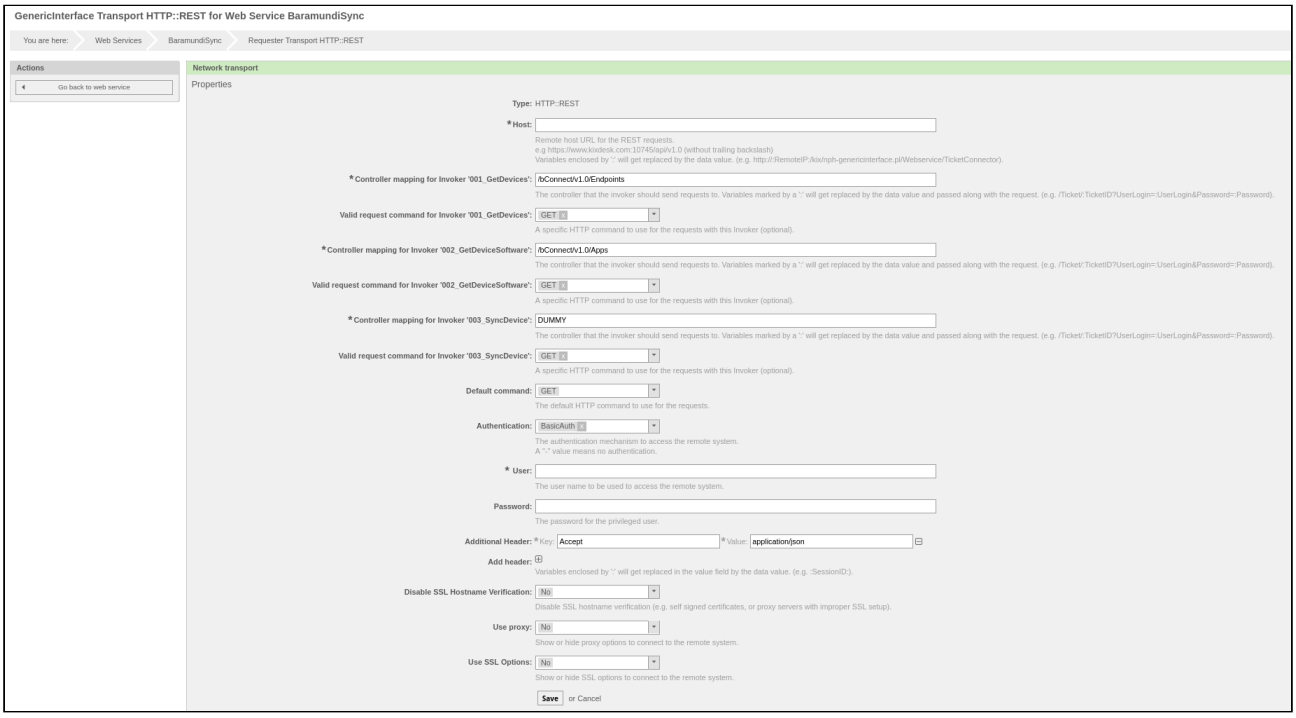


Fig.: Adapting Transport Properties

The mapping provided corresponds to certain default specifications. Here, you may also have to make adjustments to the circumstances of your environment.

- Here, select the applicable invoker in "KIX as Requester" area.

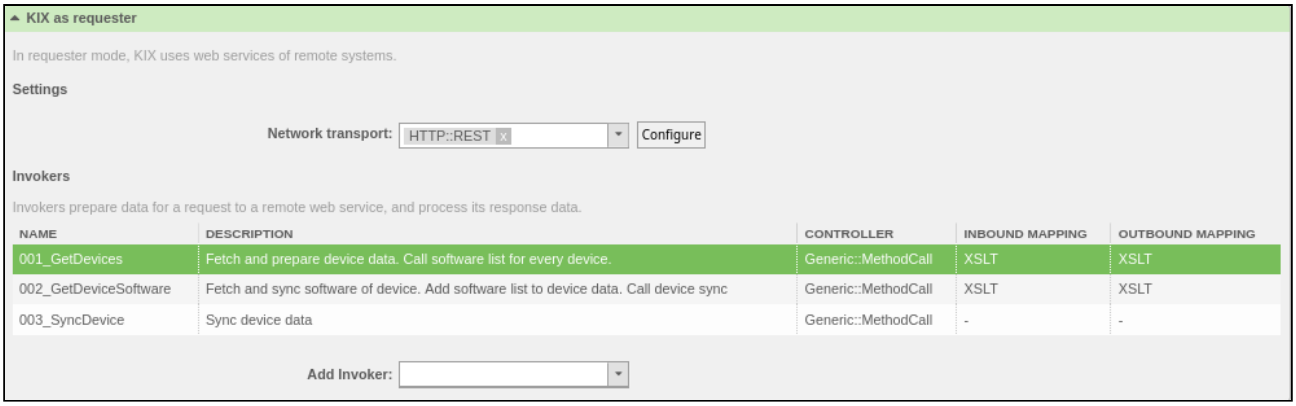


Fig.: Selecting Invoker for Mapping Adjustments

A new form opens.

- Here click "Configure" button under "Mappings" by "Incoming response data" and adapt the available mapping to your requirements.
- Finally, click "Save" button.

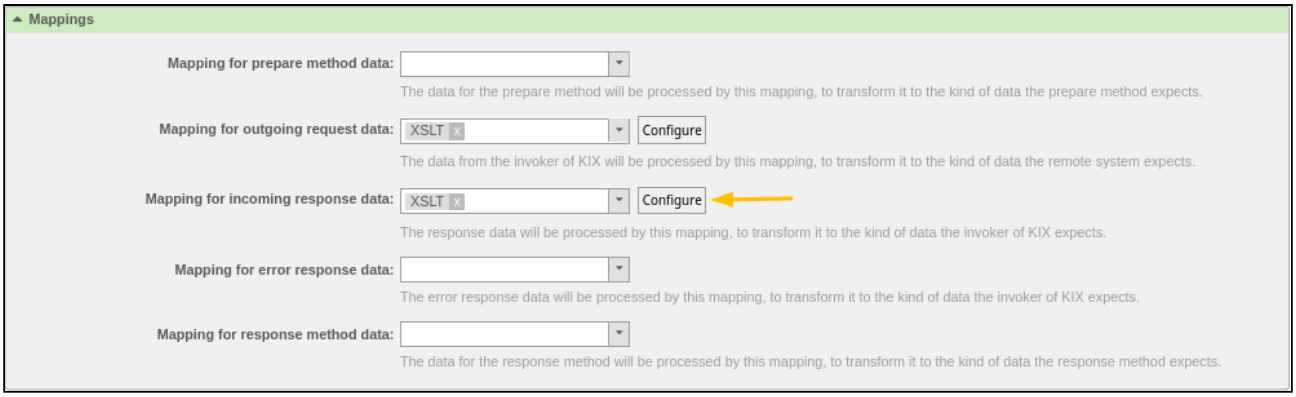


Fig.: Adapting Mappings

Proceed in the same way for the other invokers and mappings.

9.5.5 VMWareAirWatchSync

With VMWareAirWatchSync, you can synchronize devices (Device) and their app list (Software) to the KIX CMDB. For this, a web service template is provided that is inactive when delivered.

- CI class "Device" is created when KIX Connect is installed.
- CI class "Software" is used by VMWareAirWatchSync with the available default attributes.

9.5.5.1 Activating "VMWareAirWatchSync" Web Service

In the delivery state, the web service is still set to inactive. In order to activate it, carry out the following points:

- In main menu, click "Admin" entry.
- In "System Administration" area, click "Web Services" option.

The "Web Service Management - Change" area opens

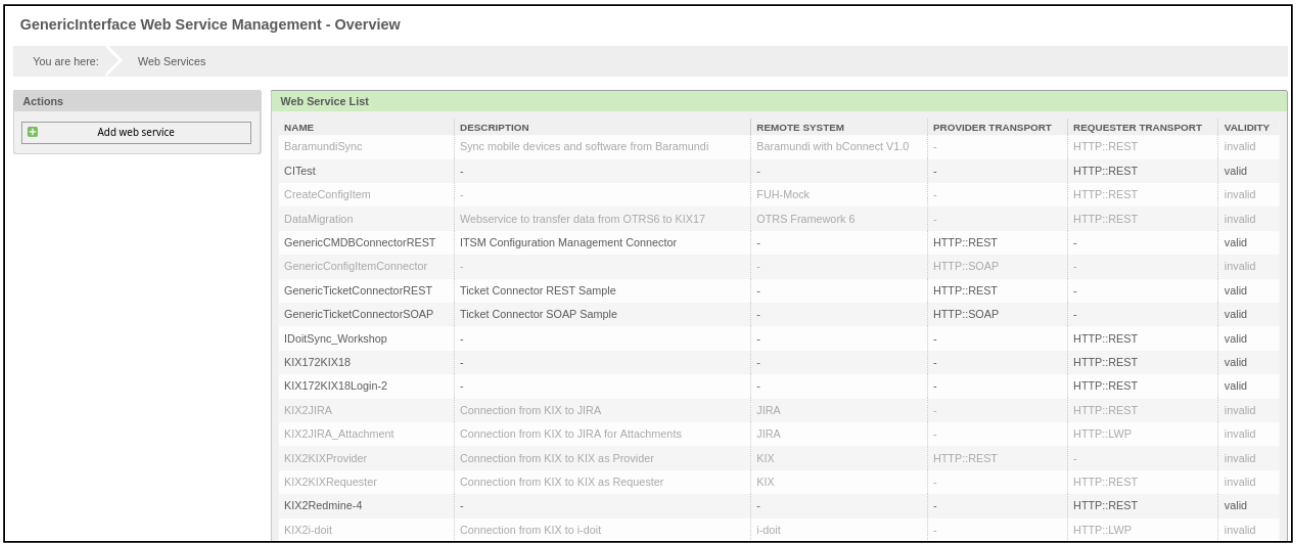


Fig.: Web Service Management - Overview

- Click entry "VMWareAirWatchSync" in the list. The "Web Service Management - Change" area opens

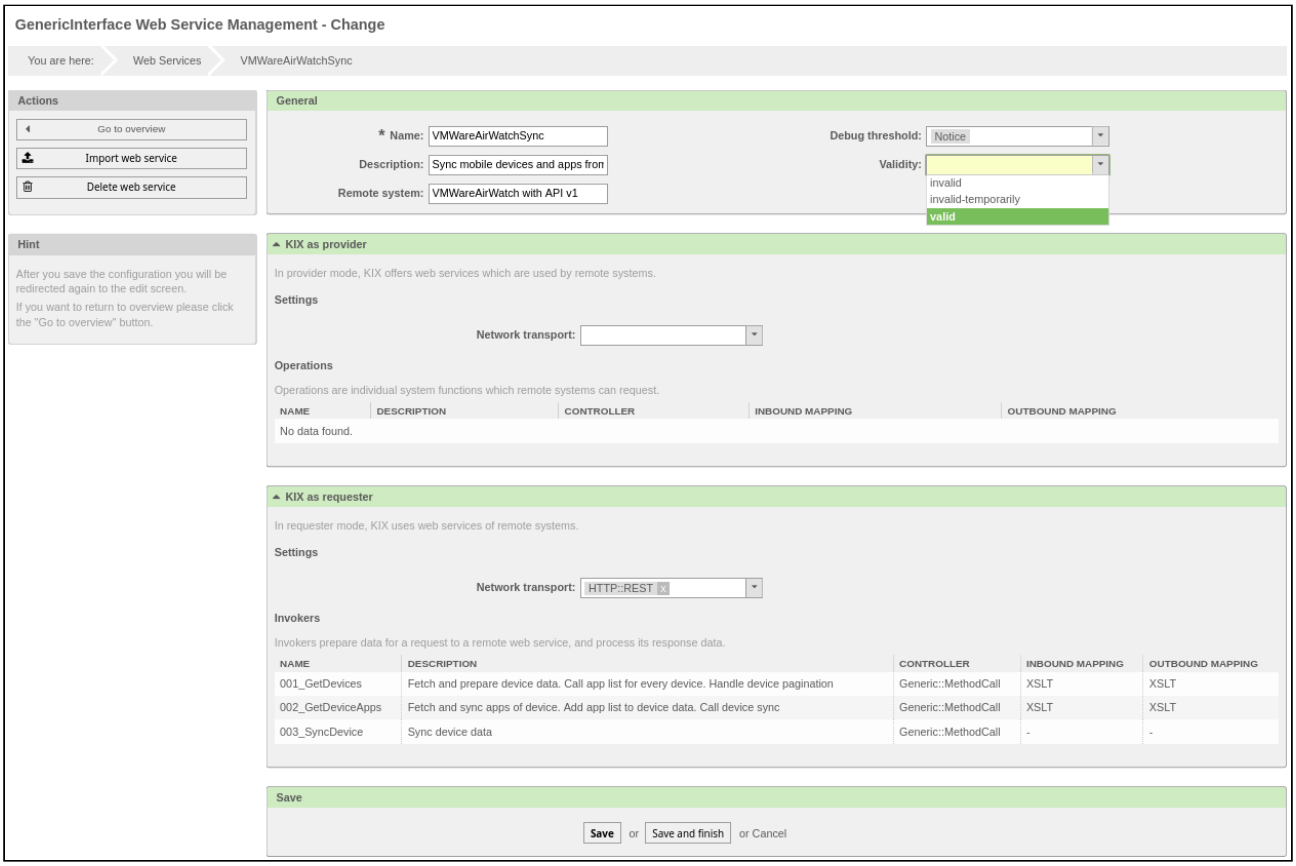


Fig.: Web Service Management - Change

- In the drop-down menu "Validity", select value "valid".
- Finally, click "Save" button.

To adapt "VMWareAirWatchSync" to your circumstances make the applicable modifications to the network transport.

- In "KIX as requester" area, click "Configure".

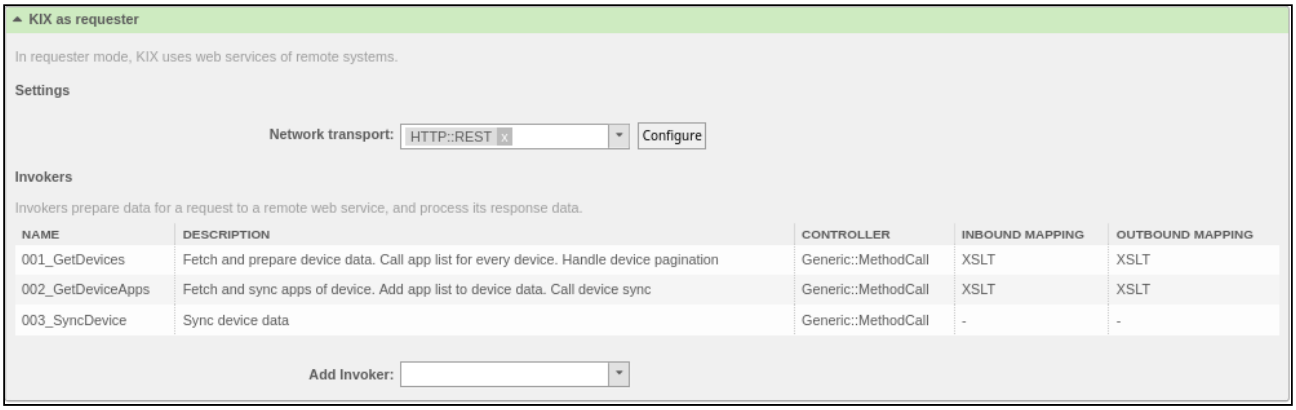


Fig.: Changing Transport Properties

- Fill out the required fields
- Finally, click "Save" button.

Fig.: Adapting Transport Properties

The mapping provided corresponds to certain default specifications. Here, you may also have to make adjustments to the circumstances of your environment.

- Here, select the applicable invoker in "KIX as Requester" area.

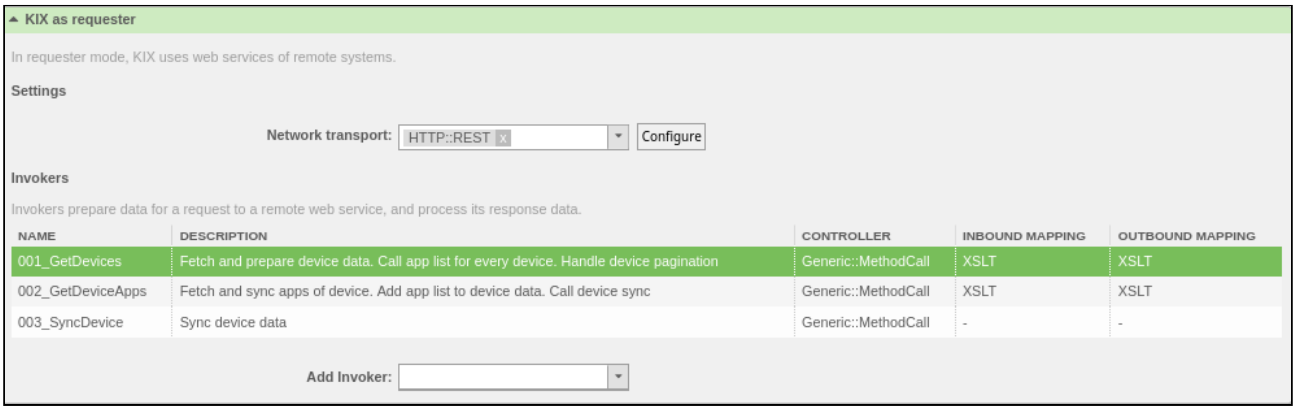


Fig.: Selecting Invoker for Mapping Adjustments

A new form opens.

- Here click "Konfigurieren (Configure)" button under "Mappings" by "eingehende Antwortdaten (Incoming response data)" and adapt the available mapping to your requirements.
- Finally, click "Save" button.

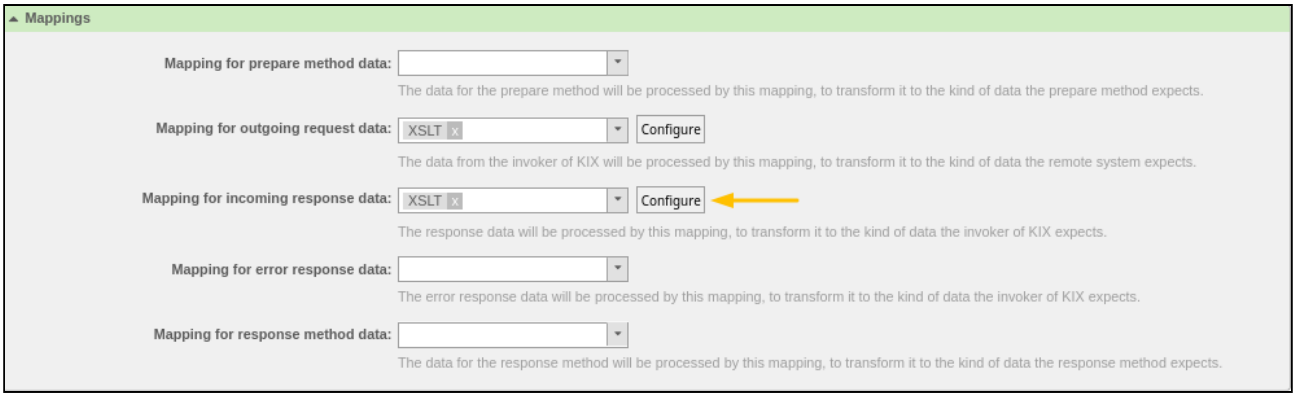


Fig.: Adapting Mappings

Proceed in the same way for the other invokers and mappings.

10 KPI Dashboard

10.1 Installation of KPI Dashboard Add-On Module

10.1.1 Prerequisites

- KIX Professional 17 – current version
- Microsoft Excel 2010 or 2013 with PowerPivot Plugin

If you would like to use a direct link from the computer on which Microsoft Excel and PowerPivot are installed and the KIX Professional database, the following prerequisite must be met:

- An ODBC driver and configured data source for the KIX Professional database.

Please note

Using an ODBC connection for the KPI database view is not recommended as this is the last flexible option to use the "KPI Dashboard" add-on. It could be necessary when you have to work with a large amount of data.

10.1.2 Installation

Install the package *"KPI-Dashboard"* using *"Paketverwaltung (Package Administration)"* (Menu *"Admin"* area *"System Administration"*) or using the command line.

Important

If you use KIX with mod_perl, you must ensure that the script /opt/kix/bin/cgi-bin /KPIDashboard.pl is not executed by mod_perl. To configure this, please add the following lines (marked in red) to the applicable section in the Apache configuration for KIX (please correct the ScriptAlias "kix" in line with your current KIX installation).

```
<IfModule mod_perl.c>
    # Setup environment and preload modules
    Perlrequire /opt/kix/scripts/apache2-perl-startup.pl

    # Reload Perl modules when changed on disk
    PerlModule Apache2::Reload
    PerlInitHandler Apache2::Reload

    # general mod_perl2 options
    <Location /otrs>
        # ErrorDocument 403 /kix/customer.pl
        ErrorDocument 403 /kix/index.pl
```

```
        SetHandler perl-script
        PerlResponseHandler ModPerl::Registry
        Options +ExecCGI
        PerlOptions +ParseHeaders
        PerlOptions +SetupEnv
        Order allow,deny
        Allow from all
    </Location>

    # mod_perl2 options for GenericInterface
    <Location /kix/nph-genericinterface.pl>
        PerlOptions -ParseHeaders
    </Location>

    # deactivate mod_perl for KPIDashboard.pl
    <Location /kix/KPIDashboard.pl>
        SetHandler cgi-script
    </Location>
</IfModule>
```

Please note

In certain older Apache versions and Linux distributions, you have to replace "perl-script" with "cgi-script" if you get an error message when loading the data stream.

10.2 Configuration of KPI Dashboard Add-On Module

Once you have installed the package, if necessary, you can set the default values in the Admin area/ SysConfig. Here, select SysConfig group "KPI-Dashboard".

After page has reloaded, select one of the subgroups "Core".

Default configuration is already contained in the package. You can adapt this in accordance with your wishes. The SysConfig settings are explained in the following sections.

Please note

If you have changed one of the settings and are using prepared KPI data, you have to run the KIX console command

- Maint::KPIDashboard::Rebuild

to rebuild the entire KPI database.

- KPI Area – Years Analyzed (see page 153)
- Default Dashboard File (see page 153)
- State Type Restriction (see page 153)
- Archive Flag Restriction (see page 153)
- Ignored Queues (see page 154)
- Ignored Ticket Types (see page 154)
- Ignored Services (see page 154)
- Ignored SLAs (see page 155)
- Escalation Base (see page 155)
- Escalation Time Base (see page 156)
- Data Stream – Directory for Data Preparation (see page 156)
- Data Stream – Dynamic Fields (see page 156)
- List of Available Ticket Attributes (see page 156)
- List of Available KPI Attributes (see page 157)
- List of Attributes Selected by Default (see page 157)
- Database View (see page 157)
- List of Available Database View Attributes (see page 158)
- Symbols for the OutOfSLA Percentage (see page 158)
- Settings for Elements in Online KPI Dashboard (see page 159)
- Explanation of Default KPI Data (see page 159)

10.2.1 KPI Area – Years Analyzed

- SysConfig setting: KPIDashboard###ShownYears

Using this setting, you can define the number of years to be analyzed for the KPI report.

Default value "3" means that the current year and the last two years are analyzed.

> **Please note**
>
> This setting can only be applied to KPI dashboards which are not based on search templates.

10.2.2 Default Dashboard File

- SysConfig setting: KPIDashboard###TemplateXML
- SysConfig setting: KPIDashboard###TemplateODBC

These settings define the location of the files for dashboard templates. If these settings are configured, it is necessary to rebuild the KPI database.

> **Attention**
>
> Please only change if absolutely necessary.

10.2.3 State Type Restriction

- SysConfig setting: KPIDashboard###StateTypes

Using this setting, you can restrict the tickets that are analyzed according to their state. The default setting is that only tickets with the *"closed"* state are analyzed. If required, you can add additional ticket states here.

> **Please note**
>
> This setting can only be applied to KPI dashboards which are not based on search templates.

10.2.4 Archive Flag Restriction

- SysConfig setting: KPIDashboard###ArchiveFlags

Using this setting, you can define which archive flag the tickets must have so they are included in the analysis for the KPI database. The default setting is for all tickets – whether archived or not – to be included in the analysis.

Please note

This setting can only be applied to KPI dashboards which are not based on search templates.

10.2.5 Ignored Queues

- SysConfig setting: KPIDashboard###IgnoredQueues

Using this setting, you can define which queues are to be ignored when tickets are analyzed for the KPI database.

This setting is a list. Each entry can contain a search pattern (regex).

Please note

This setting can only be applied to KPI dashboards which are not based on search templates.

10.2.6 Ignored Ticket Types

- SysConfig setting: KPIDashboard###IgnoredTicketTypes

Using this setting, you can define which ticket types are to be ignored when tickets are analyzed for the KPI database.

This setting is a list. Each entry can contain a search pattern (regex).

Please note

This setting can only be applied to KPI dashboards which are not based on search templates.

10.2.7 Ignored Services

- SysConfig setting: KPIDashboard###IgnoredServices

Using this setting, you can define which services are to be ignored when tickets are analyzed for the KPI database.

This setting is a list. Each entry can contain a search pattern (regex).

Please note

This setting can only be applied to KPI dashboards which are not based on search templates.

10.2.8 Ignored SLAs

- SysConfig setting: KPIDashboard###Ignored SLAs

Using this setting, you can define which SLAs are to be ignored when tickets are analyzed for the KPI database.

This setting is a list. Each entry can contain a search pattern (regex).

> **Please note**
>
> This setting can only be applied to KPI dashboards which are not based on search templates.

10.2.9 Escalation Base

- SysConfig setting: KPIDashboard###EscalationBase

This setting defines which basis should be used for escalation data when it comes to KPI analysis (e.g. solution time).

Here, depending on your configuration and ticket process, you can choose between three different values:

Option	Description
All	All tickets are analyzed. If a ticket is linked to an SLA, the escalation times are calculated based on the SLA settings. If a ticket is not linked to an SLA, the queue settings are used for analysis.
only Queue	Only tickets not linked to an SLA are analyzed. The escalation times are calculated based on the queue settings.
only SLA	Only tickets linked to an SLA are analyzed. The escalation times are calculated based on the SLA settings.

10.2.10 Escalation Time Base

- SysConfig setting: KPIDashboard###EscalationTimeBase

This setting defines the escalation times to be used to calculate KPI attribute *"OutOfSLA"*. In the default setting, the solution time is used for the calculation.

You can choose between three different values:

Option	Description
All	The *"OutofSLA"* attribute is set when one of the ticket escalation times is reached.
only First Response Time	The *"OutofSLA"* attribute is set when only the first response time is reached.
only Solution Time	The *"OutofSLA"* attribute is set when only the solution time is reached.

10.2.11 Data Stream – Directory for Data Preparation

- SysConfig setting: KPIDashboard###DataStreamPreparedDataDir

This setting defines the directory in the local file system in which the prepared data is to be saved.

10.2.12 Data Stream – Dynamic Fields

- SysConfig setting: KPIDashboard###DataStreamIncludeDynamicFields

Using this setting, you can define whether dynamic fields should be available when selecting the data stream attributes. Default value is *"Yes"*.

10.2.13 List of Available Ticket Attributes

- SysConfig setting: KPIDashboard###TicketAttributes

This setting defines the list of available attributes of the ticket data hash (including the expanded ticket attributes). All configured entries are listed in the attribute selection for dashboard administration. *"Key"* is the ticket data hash attribute and *"Content"* the name of the element in the data stream.

Note

An element in the list must start with a letter and must not contain spaces or special characters.

10.2.14 List of Available KPI Attributes

- SysConfig setting: KPIDashboard###KPIDataAttributes

This setting defines the list of available KPI attributes of the calculated KPI data hash. All configured entries are listed in the attribute selection for dashboard administration. *"Key"* is the KPI data hash attribute and *"Content"* the name of the element in the data stream. This configuration can be expanded by KPI dashboard plugins if SysConfig setting *"KPIDashboard::KPIDataAttributesExtension###..."* is used.

Note

An element in the list must start with a letter and must not contain spaces or special characters.

10.2.15 List of Attributes Selected by Default

- SysConfig setting: KPIDashboard###DefaultSelectedAttributes

Defines the list of attributes to be selected by default during the creation of a new dashboard.

Please note: The entry for *"Key"* has to be identical to the entries in the *"Key"* field in configuration options *"KPIDashboard###TicketAttributes"* and *"KPIDashboard###KPIDataAttributes"*. The entry in the *"Content"* field must be *"0"* (not selected) or *"1"* (selected).

Important

This configuration can be expanded by KPI dashboard plugins if SysConfig option *"KPIDashboard::KPIDataAttributesExtension###..."* is used.

10.2.16 Database View

- SysConfig setting: KPIDashboard###DBView

This setting defines the database view that contains the primary KPI data.

10.2.17 List of Available Database View Attributes

- SysConfig setting: KPIDashboard###DBViewAttributes

This setting contains all available attributes of the calculated KPI data in the KPI database view. All configured entries are listed in the attribute selection for dashboard administration. The entry in the *"Key"* field is the column name of the database view. The entry in the *"Content"* field represents the name of the element in the data stream. Important: An element in the list must start with a letter and must not contain spaces or special characters.

> **Please note**
>
> This setting is ignored by KPI dashboards that are based on search templates.

10.2.18 Symbols for the OutOfSLA Percentage

- SysConfig setting: KPIDashboard###OutOfSLAPercentage

This setting defines the symbols used for individual percentages in the overview table of the online KPI dashboard.

10.2.19 Settings for Elements in Online KPI Dashboard

- SysConfig settings:
  - KPIDashboardBackend###-0000-FirstSolutionRate
  - KPIDashboardBackend###-0001-IncidentDurationTopTen
  - KPIDashboardBackend###-0002-IncidentDuration
  - KPIDashboardBackend###-0003-OutOfSLA
  - KPIDashboardBackend###-0004-Overview
  - KPIDashboardBackend###-0005-QueueMove
  - KPIDashboardBackend###-0006-TicketTrendBasedOnTicketType
  - KPIDashboardBackend###-0010-Filters

These SysConfig settings describe the elements available in the online KPI dashboard, and define the settings for an element. Here, the values in the *"Key"* and *"Content"* fields each mean the following:

Key	Description of "Content" field
Block	ContentLarge or ContentSmall.
CacheTTLLocal	Cache time for the plugin in minutes.
Default Description	Defines whether the plugin is active by default or whether the user has to manually activate it.
Group	Description of the element
Module	Restricts access to the groups entered.
Title	Title shown in the online dashboard.
Type	Javascript function used to generate the display data.

10.2.20 Explanation of Default KPI Data

In the following table, you can see descriptions of the KPI attributes of the default KPI data hash. If required, the KPI data hash can be expanded by user-defined calculation modules. The table does not include the ticket hash attributes as these are fundamental KIX attributes that are not part of the KPI dashboard add-on.

KPI attribute	Description
Calendar week	Calendar week in which the ticket was created.
FirstResponseTimeTotal	Total time in minutes between the ticket being created and first reaction.
FirstResponseTimeBusiness	Value of the attribute "FirstResponseTimeTotal" minus the non-relevant times (for example, time spent waiting for a reply from the customer).
FirstTimeResolution	This attribute is set to "1" if a ticket is resolved during first contact. This means that it was created in a closed state/or there is only one outgoing message or article (email, telephone call, external note).
FirstTimeResolutionText	This is the content of the attribute "FirstTimeResolution" that a person can read, for example "yes" and "no".
Month	Month used for filtering the KPI dashboard in PowerPivot.
OutOfSLA	This value is set to "1" if the ticket has been escalated in its life cycle. The value depends on the configuration of the KPI dashboard. (See SysConfig "EscalationTimeBase").
OutOfSLAFirstResponseTimeBusiness	If the attribute "OutOfSLA" = "1", this attribute contains the number of minutes for which "FirstResponseTimeBusiness" exceeds "FirstResponse"-escalation time. The value depends on the configuration of the KPI dashboard. (See SysConfig "EscalationTimeBase")
OutOfSLASolutionTimeBusiness	If the attribute "OutOfSLA" = "1", this attribute contains the number of minutes for which "SolutionTimeBusiness" exceeds "FirstResponse"-escalation time. The value depends on the configuration of the KPI dashboard. (See SysConfig "EscalationTimeBase")

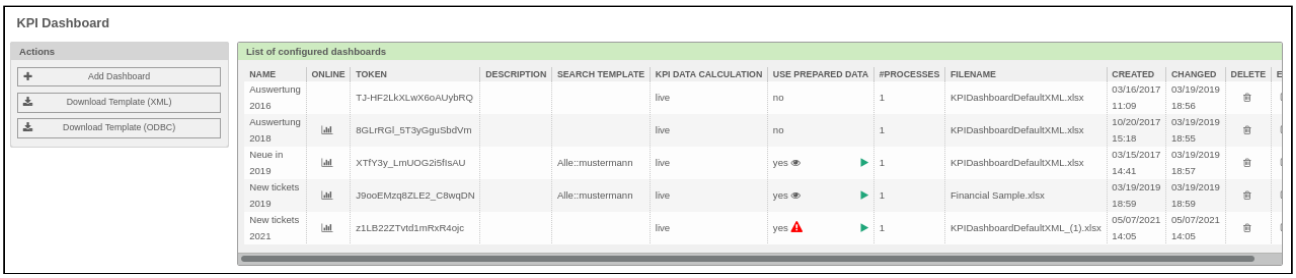
KPI attribute	Description
Queue_Level1..x	The ticket queue is divided by " :: " and each part is returned as a separate column.
QueueMoves	Number of ticket moves in different queues.
Service_Level1..x	The ticket service is divided by " :: " and each part is returned as a separate column.
SolutionTimeTotal	Total time in minutes between the ticket being created and being closed.
SolutionTimeBusiness	Value of the attribute "SolutionTimeTotal" minus the non-relevant times. (For example, time spent waiting for a reply from the customer).
TicketPriorityTranslated	Translated ticket priority for filtering the KPI dashboard in PowerPivot.
TicketStateTranslated	Translated ticket state for filtering the KPI dashboard in PowerPivot.
Year	Calendar year in which the ticket was created.

10.3 Administration of KPI Dashboard Add-On Module

If you want to create new KPI dashboards or edit existing ones, in the *"Reports"* menu, select the item *"KPI Dashboard"*.

The user must have read/writing rights for the *"KPI Dashboard"* group to create or edit new KPI dashboards.

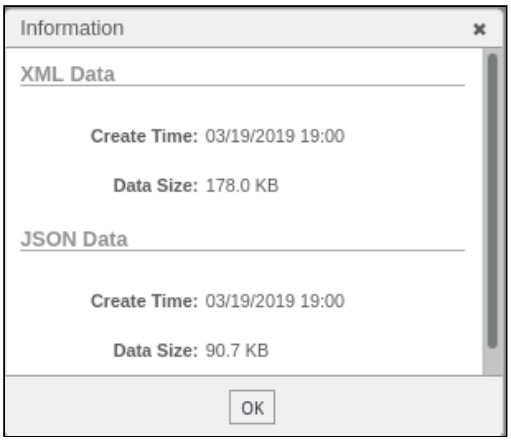
In the *"KPI Dashboard"* area, you can see a list of the configured KPI Dashboards (if KPI dashboards have already been configured). In the list, in the column *"Use prepared data"* a symbol is shown when a KPI Dashboard is configured for prepared data, and prepared data is available for this KPI dashboard. If no prepared data is available, a symbol is shown in the column instead.



You can click to see data information. If data preparation processes are running in the background, a progress display is shown with a stop symbol.

If you click the stop symbol, data preparation is stopped. It will take a few seconds to stop the process. In addition, a symbol is shown alongside the progress display on the left.

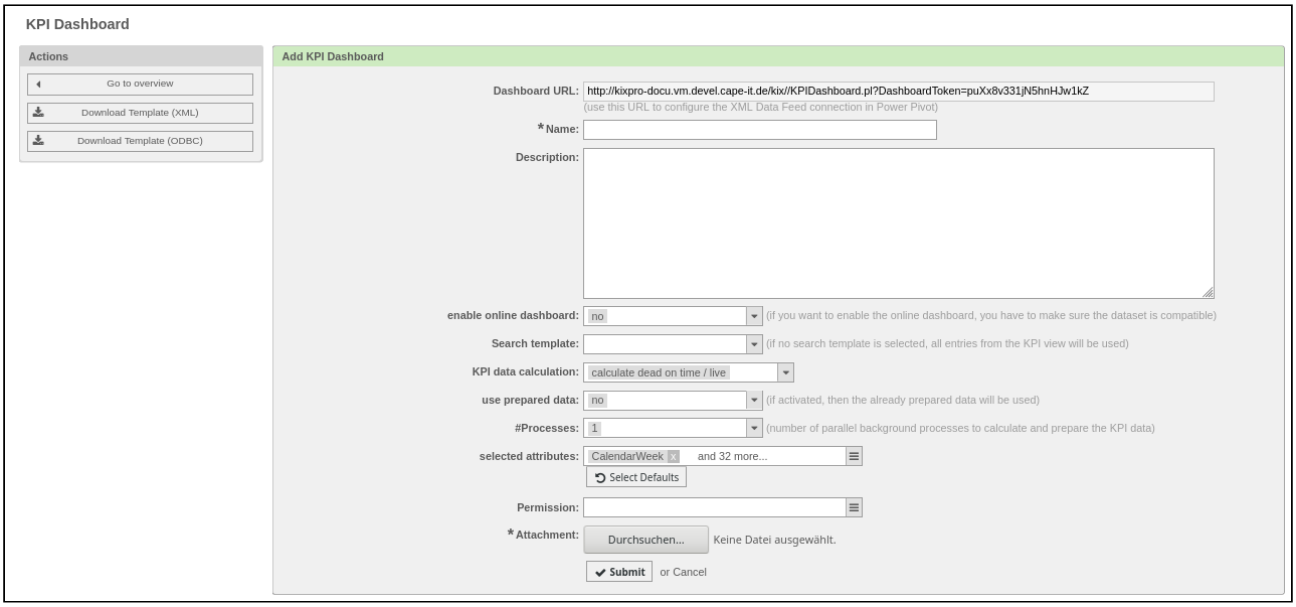
If you click the symbol, a popup displays information on each background data preparation task.



Please note: If errors occur while creating or editing KPI dashboards, messages appear in the information bar below the menu bar.

10.3.1 Creating New KPI Dashboards

If you create a new dashboard using the "Add Dashboard" button, the following widget is shown:



The widget contains the following options:

Term	Explanation
Dashboard URL	This is the URL of the XML data feed connection.
Name*	Name of KPI-Dashboard.
Description	Descriptions for the KPI dashboard can optionally be entered here.
Enable Online Dashboard	Here you can select whether an online version of this KPI dashboard should be made available. Please note: If you want to make the online dashboard available, you must ensure that your online configuration is compatible with the dataset of this dashboard.

Term	Explanation
Search Template	This option makes it possible to select one of your own ticket search profiles for searching for relevant tickets. If no search profile can be assigned, the KPI database view is used.
KPI Data Calculation	Here you can specify whether live calculation of KPI data should be used, or whether pre-calculated data should be used. Please note: Changing this option also changes the available attributes in the attribute list. If you select the option "use pre-calculated data from DBView", the calculated values are removed, and the values from the KPI database view used instead.
Use Prepared Data	Using this option, you can define whether prepared data should be used. (This requires certain automated jobs to regularly prepare the necessary data). If "no" is selected for this option, the data is generated immediately, which usually takes more time. The data is always then up to date, however.
#Process	Here you can define how many parallel processes should be used to calculated the KPI data. The more processes, the higher the system load on the KIX server. If the number of tickets to be analyzed is relatively small, always only use 1 process. In the event of higher numbers of tickets, more processes are recommended. Please note: This option always generates the selected number of parallel processes, irrespective of the actual physical CPU number of your KIX server. If the number of processes is higher than the CPU number, this will have a substantial impact on the system performance during the calculation.
Selected Attributes	Depending on the setting for "Search template" and "KPI data calculation" and depending on the settings in SysConfig, here you can select the attributes that should be contained in the data. Please note: When creating a new KPI dashboard, certain attributes are pre-selected by the system. These attributes are defined in SysConfig. You can revert back to the default setting at any time by clicking the "Select Defaults" button.

Term	Explanation
Permission	For users to be able to access a KPI dashboard, you have to select one or more groups here. Users with read/write rights in these groups can view and edit KPI dashboards. Users with read rights can only view KPI dashboards.
Attachment*	Here you can upload the XLS file that contains the PowerPivot dashboard. The data connection in line with your type of access to KIX data (XML data feed or ODBC connection) must already be configured within this XLS file.

Fields marked with * must be filled out.

10.3.2 Configuring KPI Data Source in PowerPivot

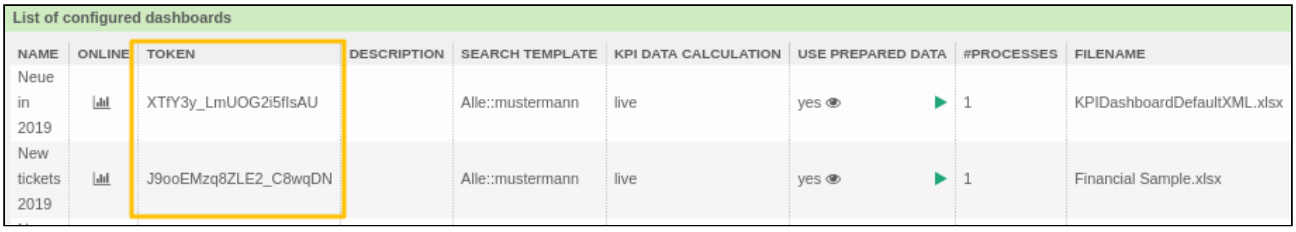
In PowerPivot you have to configure a data source in order to be able to display the KPI data. Here, you can either use an XML data feed or an ODBC connection.

XML data feed:

This is the preferred data source for use as no database connections, open ports, or extended database tables have to be configured to transfer more attributes.
To connect PowerPivot with a dashboard XML data feed, you have to specify the data source in PowerPivot. Use the following URL to incorporate the XML data feed in PowerPivot:

```
http://<FQDN KIX Server>/kix/KPIDashboard.pl?DashboardToken=<Token>
```

You can find the token to use in the list of configured dashboards. A random token is generated for each new KPI dashboard. The token is used to identify the specific KPI dashboard in the URL.



If you use HTTPS links, you have to change the URL as appropriate.

ODBC:

The ODBC connection can be used when XML data feeds cannot be used. This enables you to transfer data from the KIX database directly to PowerPivot. This provides certain advantages when it comes to performance. A disadvantage, however, is that you generally have to open firewall ports to access the DBMS. You can find information on how to create an ODBC data source in your Excel documentation.

10.3.3 Preparing Data Feeds

If you have configured the KPI dashboards using the option "use prepared data", you must ensure that the necessary data feeds are prepared in the background. An XML data feed is necessary to use the data in PowerPivot. A JSON data feed is necessary to use the data in the online KPI dashboard. There are three possibilities to prepare the data:

- Preparation of the data in the KIX GUI
- Preparation of the data using KIX console scripts
- Preparation of the data using cron job

10.3.4 Preparing Data Feed using KIX GUI

If you have permission to edit a KPI dashboard and the settings are configured so that prepared data is used, a  symbol is shown in the list of configured KPI dashboards. If you click the 
symbol, data preparation is started in the background.

Once data preparation has been started, in the list you can see a progress display with a  symbol. If you click the  symbol, data preparation is stopped. It takes a while for the process to stop.

In addition, a  symbol is shown alongside the progress display. If you click the  symbol, a popup displays information on each background data preparation task.

10.3.5 Preparing Data Feed using KIX Console Script

Here, use the console command:

```
Maint::KPIDashboard::Prepare
```

If you call up this command without parameters, an XML data feed and a JSON data feed is created for each KPI dashboard that requires preparation. The same number of parallel processes are used as specified for the creation of the KPI dashboard.

Optionally, you can specify the parameter "--type" with the values "XML" or "JSON" in order to prepare the desired data format.

10.3.6 Preparing Data Feed using Cron Jobs

If you regularly prepare data, you can define a cron job for this purpose. The necessary cron job template is already installed in the var/cron directory.

10.4 Data Plugins for KPI Dashboard

The KPI dashboard core enables you to expand and/or overwrite the available KPI data using plugin modules.

These plugin modules have to be registered in SysConfig (SysConfig key "KPIDashboard :: CustomModule"). The plugin modules must be located in the path /opt/kix/Kernel/System/KPIDashboard/. The plugin module must only provide two methods: "new" and "Run".

The method "Run" receives two parameters:

"TicketID" and "Data"

and must return a hash with data attribute keys and values. The parameter "Data" contains the KPI data already calculated for a ticket. The list of the new attributes of a plugin module can be made available using SysConfig add-ons (see SysConfig keys "KPIDashboard###KPIDataAttributes" and "KPIDashboard###DefaultSelectedAttributes").

10.5 Online KPI Dashboard

The online KPI dashboard provides the possibility to display standard diagrams without using PowerPivot. It works like the standard ticket dashboard. Each dashboard element is an individual widget that can be activated/deactivated in SysConfig group "Frontend::Agent::OnlineDashboard". Agents can also switch on/off the widget for themselves.

10.5.1 Opening Online KPI Dashboard

The online KPI dashboard is only available if it has been configured applicably in the "Add KPI Dashboard" widget.

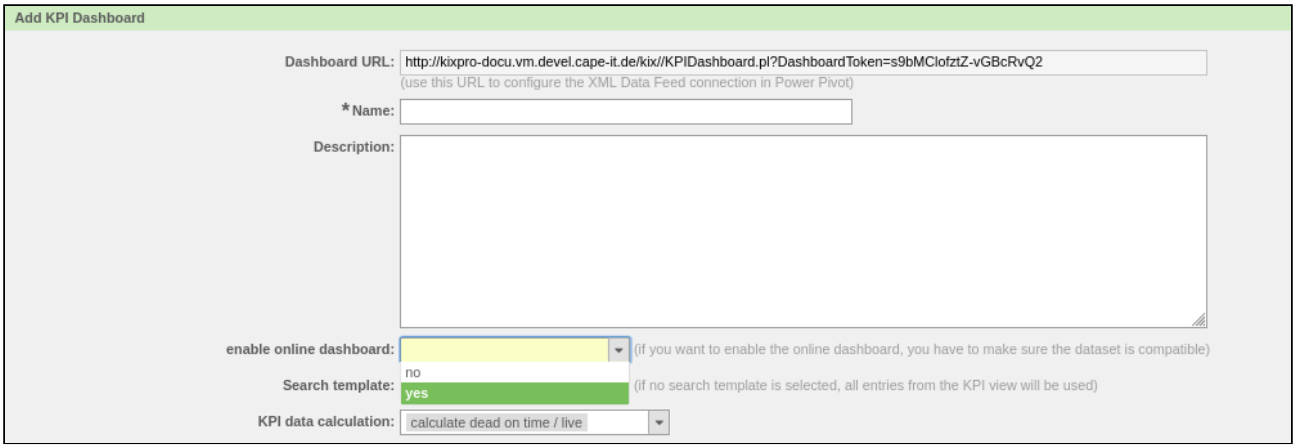


Fig.: "Add KPI Dashboard" Widget

If an online KPI dashboard is available, a  symbol is shown in the list of available dashboards. Clicking this symbol opens the online KPI dashboard.

A new browser tab opens and the online KPI dashboard opens. A small popup informs the user of this. It is displayed for as long as data processing is running.

If the online KPI dashboard uses prepared data, the entire JSON data feed is loaded into the browser. If the online KPI dashboard is configured so that no prepared data is used, data calculation starts. The JSON data feed is loaded into the browser afterward.

Please note

As the complete JSON data feed is loaded into the browser, it can take a few minutes for the loading of the data to finish. The memory requirements of the browser are also increased significantly.

10.5.2 Settings for the Online KPI Dashboard

In the online KPI dashboard, the data is shown in widgets. In the left area of the browser you can find the "Settings" widget. Here there is a list of all the widgets activated by the KIX administrator.

Here, the agents can hide or show widgets as they prefer and save the settings. Each widget has a predefined position on the screen, but can be moved using drag&drop. The position of the widgets is saved when you log out of KIX.

10.5.3 Filter for the Online KPI Dashboard

If active, the "Filter" widget is shown on the left side of the browser. Using this, the data can be filtered and displayed in the online KPI dashboard.

SysConfig setting "KPIDashboardBackend###010-Filters" is used to define which attributes are filtered. You can find the filter attributes in the "Filters" key.

The values shown in the filter are taken over live from the JSON data feed and represent all the possible values in the current data feed. All active filters are linked to each other with a logical AND.

No filters are set once the online KPI dashboard has loaded. To filter the data, select the applicable values in the "Filters" widget and click "Apply".

11 Maintenance Planner

11.1 Installation of Maintenance Planner Add-On Module

11.1.1 Prerequisites

KIX Professional 17 – current version

11.1.2 Package Installation

Install the "Maintenance Planner" package using the package manager of the operating system or the command line: kixpro-maintenanceplan.

If the class "Maintenance plan" is not available, you can find the class definition under "<KIXHome>/var/packagesetup/InitialCIClassDefinitions/" .

11.1.3 New Access Control Lists (ACL)

During installation, three new ACLs are created in the system.

"500_TicketTypeMaintenance_OnlyInAgentfrontend"	Means ticket type "Maintenance Task" is not shown in customer frontend.
"500_TicketTypeMaintenance_ActionAgentTicketX"	13 ticket actions (menu items in ticket overview). Exactly which these are can be found in the ACL directly: Click Admin → ACL Management→ ACL "500_TicketTypeMaintenance_ActionAgentTicketX".
"500_TicketTypeMaintenance_ActionAgentTicketMaintenance"	If a ticket is not of the type "Maintenance Task", the maintenance tab is not shown and action "Maintenance Documentation" is not available.

11.1.4 Adapting the CI Classes

Adapt the CI classes for the Maintenance Planner as applicable: "<KIXHome>/var/packagesetup/InitialCIClassDefinitions/MaintenanceCI.def".

11.1.5 New Ticket Type

During installation, a new ticket type "Maintenance Task" is created. It is used to identify maintenance tickets.

11.1.6 New Dynamic Fields

The dynamic fields that are newly available are relevant for maintenance tickets that are automatically created when the maintenance time has been reached.

Field name	Explanation
MaintenancePlan	Contains the ID of the relevant MaintenancePlan ConfigItem.
MaintenanceDate	Contains the date of the maintenance.
MaintenanceCI	Contains the ID of the relevant ConfigItem that is to undergo maintenance.
MaintenanceCIClass	Contains the class of the relevant ConfigItem that is to undergo maintenance.
MaintenancePlanTime	Contains the planned time for this maintenance.
MaintenanceCILocation	Contains the location ID of the relevant ConfigItem that is to undergo maintenance.
MaintenanceCostCenter	Contains the applicable cost center ID of the relevant ConfigItem that is to undergo maintenance.
MaintenanceCIContact	Contains the ID of the contact of the relevant ConfigItem that is to undergo maintenance.
MaintenanceCIType	Contains the type of the relevant ConfigItem that is to undergo maintenance.
MaintenanceSupplier	Contains the customer/supplier performing the maintenance.
MaintenanceDueDate	Contains the due date of time-based maintenance.
MaintenanceThreshold	Contains the threshold value of threshold-based maintenance.

Field name	Explanation
MaintenanceThresholdAttribute	Contains the attribute name with the current operating count value of the ConfigItem to undergo maintenance.
MaintenanceReachedValue	Contains the current operating count value of the ConfigItem to undergo maintenance.
MaintenanceSupplier	Contains the customer/supplier performing the maintenance.
MaintenanceDueDate	Contains the due date of time-based maintenance.
MaintenanceThreshold	Contains the threshold value of threshold-based maintenance.
MaintenanceThresholdAttribute	Contains the attribute name with the current operating count value of the ConfigItem to undergo maintenance.
MaintenanceReachedValue	Contains the current operating count value of the ConfigItem to undergo maintenance.
MaintenanceType	Contains the type of the relevant ConfigItem that is to undergo maintenance.
MaintenanceCosts	Contains the costs of the relevant ConfigItem that is to undergo maintenance.
RepairCosts	Contains the repair costs of the relevant ConfigItem that is to undergo maintenance.
MaintenancePlannedStart	Contains the planned start time.
MaintenancePlannedEnd	Contains the planned end time.
MaintenanceActualStart	Contains the actual start time.
MaintenanceActualEnd	Contains the actual end time.

Field name	Explanation
MaintenanceType	Contains the type of the relevant ConfigItem that is to undergo maintenance.
MaintenanceCosts	Contains the costs of the relevant ConfigItem that is to undergo maintenance.
RepairCosts	Contains the repair costs of the relevant ConfigItem that is to undergo maintenance.
MaintenancePlannedStart	Contains the planned start time.
MaintenancePlannedEnd	Contains the planned end time.
MaintenanceActualStart	Contains the actual start time.
MaintenanceActualEnd	Contains the actual end time.

11.2 Configuration of Maintenance Planner Add-On Module

To configure settings for this function, select "KIX Pro" group in SysConfig.

After page has reloaded, select one of the subgroups:

- Core::CMDBMaintenanceCheckScheduler
- Core::ITSMConfigItem
- Core::Ticket
- Frontend::Agent::KSBSimilarTicketList
- Frontend::Agent::Ticket::ViewMaintenanceDocumentation
- Frontend::Agent::Ticket::ViewTicketZoomTabMaintenance

- Classes for Time-Based Maintenance (see page 176)
- Classes for Threshold-Based Maintenance (see page 177)
- Relevant Deployment States (see page 177)
- Checking Threshold-Based Maintenance (see page 177)
- Link Types (see page 178)
- Replacing the Current ConfigItem Version (see page 178)
- Default Ticket Data (see page 178)
- Ticket Attributes – ConfigItem-Attribute Mapping (see page 178)
- Default Article Data (see page 179)
- Scheduler Registration (see page 179)
- Event for Validity Check (see page 179)
- Validity Check – Parameters for Maintenance Planner (see page 180)
- Validity Check – Parameters for Maintained ConfigItems (see page 180)
- Event for Checking Value Update for Threshold-Based Maintenance (see page 181)
- Writing Back Ticket Data to a ConfigItem (see page 181)
- ConfigItem Attributes – Mapping of Dynamic Fields (see page 181)
- Parameters for Searching for Similar Tickets (see page 181)
- Permission for "Change" Popup (see page 181)
- Setting Required Lock (see page 181)
- Creating Maintenance Tickets (see page 181)

11.2.1 Classes for Time-Based Maintenance

- SysConfig setting: CMDBMaintenanceCheckScheduler::CIClassesTimeBased

Using this setting, you define the ConfigItem classes relevant for time-based maintenance. The entry in the field "Key" is the class name. The entry in the field "Content" can be "1" or "0". With the value "1",

"NextCheckDateReminder" is used to check whether a maintenance ticket has to be created. With the value "0", "NextCheckDate" is used to check whether a maintenance ticket has to be created.

Please note

This setting is required. This setting is initially delivered empty, so it is essential it is completed before first use.

11.2.2 Classes for Threshold-Based Maintenance

- SysConfig setting: CMDBMaintenanceCheckScheduler::CIClassesThresholdBased

Using this setting, you define the ConfigItem classes relevant for threshold-based maintenance. The entry in the field "Key" is the class name. The entry in the field "Content" can be "1" or "0". With the value "1", "NextCheckDateReminder" is used to check whether a maintenance ticket has to be created. With the value "0", "NextCheckDate" is used to check whether a maintenance ticket has to be created.

Please note

This setting is required. This setting is initially delivered empty, so it is essential it is completed before first use.

11.2.3 Relevant Deployment States

- SysConfig setting: CMDBMaintenanceCheckScheduler::CICassesDeploymentStates

Using this setting, you define which deployment states are relevant for the respective class in order to check whether a maintenance ticket has to be created. The entry in the field "Key" is the class name. The entry in the field "Content" is the deployment state. Here you can specify several deployment states separated by commas or semicolons.

Note

Please note: This setting is also required. If no state is saved here, the CIs are not checked.
Attention: The English name of the deployment state must be entered.

11.2.4 Checking Threshold-Based Maintenance

- SysConfig setting: CMDBMaintenanceCheckScheduler::CheckThresholdBased

Using this setting, you define whether threshold-based maintenance should also be checked by the Scheduler. This can result in multiple tickets for the same maintenance if the check is run again. Default value is "Yes".

11.2.5 Link Types

- SysConfig setting: CMDBMaintenanceCheckScheduler::TicketLinkType

Using this setting, you define the link type between a maintenance ticket and the applicable ConfigItem. Default value is "RelevantTo".

> **Please note**
>
> This setting is required.

11.2.6 Replacing the Current ConfigItem Version

- SysConfig setting: CMDBMaintenanceCheckScheduler::ReplaceCIVersion

Using this setting, you define whether a change to the value for "Datum letzte Wartung (Date of last maintenance)" or "Nächste Prüfung (NextCheck)" replaces the current version of a ConfigItem ("Yes") or a new version is created ("No"). Default value is "Yes".

11.2.7 Default Ticket Data

- SysConfig setting: CMDBMaintenanceCheckScheduler::NewTicket###TicketData

Using this setting, you define the ticket data for new maintenance tickets. The entry in the field "Key" can be a ConfigItem class followed by ":::" and a ticket attribute. The setting also serves as a fallback.

> **Please note**
>
> This setting is required.

11.2.8 Ticket Attributes – ConfigItem-Attribute Mapping

- SysConfig setting: CMDBMaintenanceCheckScheduler::NewTicket###TicketDataCIAttributeMapping

Using this setting, you define the mapping as to which ticket attribute (entry in field "Key") receives its value from which ConfigItem attribute (entry in field "Content"). The entry in the field "Key" can be a ConfigItem

class followed by ":::" and a ticket attribute or a dynamic field. For dynamic fields, please use the syntax "DynamicField_Fieldname". The entry in the field "Content" must start with "Maintenance:::", followed by an attribute, if the attribute is a sub-attribute of a maintenance attribute in the relevant ConfigItem classes.

Please note

This setting is required. It is recommended that you use a ConfigItem reference attribute (e.g. "ServiceReference"), if the relevant ticket attribute is not a dynamic field. Only dynamic fields of the type "Ticket" are possible.

11.2.9 Default Article Data

- SysConfig setting: CMDBMaintenanceCheckScheduler::NewTicket###ArticleData

Using this setting, you define the article data for a newly created maintenance ticket. The setting also serves as a fallback. The entry in the field "Key" can be a ConfigItem class followed by ":::" and an article attribute.

Please note

This setting is required.

11.2.10 Scheduler Registration

- SysConfig setting:
  Daemon::SchedulerCronTaskManager::Task###CMDBMaintenanceCheckScheduler

This setting is the registration of the Scheduler. When the Scheduler is running, all maintenance of relevant ConfigItems is checked and maintenance tickets created if necessary. The value in the field "Content" for the key "Scheduler" is a cron notation.

Please note

This setting is required. Please only deactivate the setting if absolutely necessary. It is also necessary for the daemon to be running.

11.2.11 Event for Validity Check

- SysConfig setting: ITSMConfigItem::EventModulePre###100-CheckAttributes

This setting is the registration for a ConfigItem event module that checks the content of ConfigItem attributes BEFORE a new version of the ConfigItem is created. Use the following settings "CheckCIAttributes###CIAttributeValuesMaintenancePlan" and "CheckCIAttributes###CIAttributeValuesMaintenanceCI" to determine the parameters for the check. This event also ensures that each maintenance plan is only used once in a ConfigItem to be maintained.

11.2.12 Validity Check – Parameters for Maintenance Planner

- SysConfig setting: CheckAttributes###CIAttributeValuesMaintenancePlan

Using this setting, you define the ConfigItem attributes of ConfigItem class "MaintenancePlan" for the validity check. The entry in the field "Key" is the attribute followed by ":::" and a specific value of the applicable attribute. The entries in the field "Content" are other attributes that are obligatory when the special value is used. If the check provides a negative result, an error message is displayed.

Please note

This setting is required. The default setting specifies that, for time-based maintenance, the attribute "Interval" must have a value. The default setting also specifies that the attribute "Contact" must have content when the customer user type is set to "fixed".

11.2.13 Validity Check – Parameters for Maintained ConfigItems

- SysConfig setting: CheckCIAttributes###CIAttributeValuesMaintenanceCI

Using this setting, you define the ConfigItem attributes of the given ConfigItem class for the validity check. The entry in the field "Key" is the class followed by "###" and a maintenance sub-attribute followed by ":::" and a specific value of the respective attribute. The entries in the field "Content" are other attributes that are obligatory when the special value is used. Use "Maintenance:::" in front of the obligatory attribute to identify it as a sub-attribute of the respective ConfigItem class. If a class is not specified, the specific entry is used for "CMDBMaintenanceCheckScheduler::CIClasses...". If a ConfigItem does not have a value for the specified sub-attribute, the applicable attribute of the relevant maintenance plan ConfigItem is checked. If this also fails, an error message appears.

Please note

This setting is required. The default setting specifies that, for the maintenance currently being considered, the attributes "Interval" and "NextCheckDate" must have a value if "time-based" has been specified for the maintenance type. If "threshold-based" has been specified as the maintenance type, the attributes "Maintenance" and "NextCheck" must have content.

11.2.14 Event for Checking Value Update for Threshold-Based Maintenance

- SysConfig setting: ITSMConfigItem::EventModulePost###100-ThresholdCheck

This is the registration of a ConfigItem event module for threshold-based maintenance. The module checks whether a maintenance ticket should be created for a ConfigItem for threshold-based maintenance. The relevant options from "Core::CMDBMaintenanceCheckScheduler" are used.

11.2.15 Writing Back Ticket Data to a ConfigItem

- SysConfig setting: Ticket::EventModulePost###600-WriteMaintenanceTicketDataIntoCI

11.2.16 ConfigItem Attributes – Mapping of Dynamic Fields

- SysConfig setting: UpdateMaintenanceCIData###CIAttributeMapping

11.2.17 Parameters for Searching for Similar Tickets

- SysConfig setting: KIXSideBarSimilarTickets###000 (010, 020)

11.2.18 Permission for "Change" Popup

- SysConfig setting: Ticket::Frontend::AgentTicketMaintenance###Permission

11.2.19 Setting Required Lock

- SysConfig setting: Ticket::Frontend::AgentTicketMaintenance###RequiredLock

11.2.20 Creating Maintenance Tickets

Checking whether maintenance tickets have to be created is performed either by the daemon in the background or via this console script:

```
sudo -u www-data bin/kix.Console.pl Admin::MaintenancePlan::Check
```

11.3 Use of Maintenance Planner

Here you will find different approaches to make using the module easier for you.

11.3.1 Adapting CI Classes for the Maintenance Planner

To save a maintenance plan for a ConfigItem, the CI class of the ConfigItem has to be adapted as applicable beforehand (path in installation "<KIXHome>/var/packagesetup/ InitialCIClassDefinitions/ MaintenanceCI.def").

11.3.1.1 Adapting for Time-Based Maintenance

If you would like to create time-based maintenance, adapt the code as follows:

```
[
    {
        Key        => 'AssignedCostCenter',
        Name       => 'Assigned Cost Center',
        Searchable => 1,
        Input => {
            Type                          => 'CIClassReference',
            ReferencedCIClassName         => 'Cost Center',
            ReferencedCIClassLinkType     => 'RelevantTo',
            ReferencedCIClassLinkDirection => '',
            Required => 0,
        },
    },
    {
        Key        => 'CILocation',
        Name       => 'Asset Location',
        Searchable => 1,
        Input => {
            Type                          => 'CIClassReference',
            ReferencedCIClassName         => 'Location',
            ReferencedCIClassLinkType     => 'RelevantTo',
            ReferencedCIClassLinkDirection => '',
            Required => 0,
        },
    },
    {
        Key        => 'ResponsibleUser',
        Name       => 'Responsible user',
        Searchable => 1,
        Input => {
            Type      => 'Customer',
        },
```

```
    },
    {
        Key        => 'Maintenance',
        Name       => 'Maintenance',
        Searchable => 1,
        Input => {
            Type                           => 'CIClassReference',
            ReferencedCIClassName          => 'MaintenancePlan',
            ReferencedCIClassLinkType      => 'Includes',
            ReferencedCIClassLinkDirection => 'Reverse',
            Required                       => 1,
        },
        Sub => [
            {
                Key   => 'MaintenanceType',
                Name  => 'Maintenance type',
                Input => {
                    Type        => 'GeneralCatalog',
                    Class       => 'ITSM::ConfigItem::Maintenance::Type',
                    Translation => 1,
                    Required    => 0,
                },
            },
            {
                Key   => 'TimeBased',
                Name  => 'Time based',
                Input => {
                    Type     => 'Dummy',
                    Required => 0,
                },
                Sub => [
                    {
                        Key        => 'Interval',
                        Name       => 'Interval',
                        Searchable => 1,
                        Input      => {
                            Type => 'GeneralCatalog',
                            Class => 'ITSM::ConfigItem::Maintenance::Interval',
                            Translation => 1,
                            Required => 0,
                        },
                    },
                    {
                        Key => 'NextCheckDate',
                        Name => 'Next check date',
                        Searchable => 1,
                        Input => {
                            Type => 'Date',
                            Required => 1,
                        },
                        CountMin => 0,
                        CountMax => 1,
```

```
                CountDefault => 1,
            },
            {
                Key => 'NextCheckDateReminder',
                Name => 'Next check date reminder',
                Searchable => 1,
                Input => {
                    Type => 'Date',
                    Required => 1,
                },
                CountMin => 0,
                CountMax => 1,
                CountDefault => 1,
            },
        ],
        CountMin => 0,
        CountDefault => 0,
        CountMax => 1,
    },
    {
        Key        => 'TicketTemplate',
        Name       => 'Ticket-Template',
        Searchable => 1,
        Input         => {
            Type      => 'TicketTemplateReference',
            Required  => 0,
        },
        CountMin     => 0,
        CountDefault => 0,
        CountMax     => 1,
    },
    {
        Key        => 'MaintenanceSupplier',
        Name       => 'Maintenance supplier',
        Input         => {
            Type      => 'CIACCustomerCompany',
            Required  => 0,
        },
        CountMin     => 0,
        CountMax     => 1,
    },
    {
        Key        => 'PlannedMaintenanceTime',
        Name       => 'Planned maintenance time (min)',
        Searchable => 1,
        Input         => {
            Type      => 'Text',
        },
        CountMin     => 0,
        CountMax     => 1,
        CountDefault => 0,
    },
```

```
            {
                Key => 'LastMaintenanceDate',
                Name => 'Last maintenance date',
                Searchable => 1,
                Input => {
                    Type => 'Date',
                    Required => 1,
                },
                CountMin => 0,
                CountMax => 1,
                CountDefault => 0,
            },
            {
                Key => 'MaintenanceNote',
                Name => 'Maintenance note',
                Searchable => 1,
                Input => {
                    Type => 'TextArea',
                    Required => 1,
                },
                CountMin => 0,
                CountMax => 1,
                CountDefault => 0,
            },
        ],
        CountMin => 0,
        CountMax => 10,
    },
];
```

11.3.1.2 Adapting for Threshold-Based Maintenance

If you would like to create threshold-based maintenance, adapt the code as follows:

```
[
    {
        Key        => 'AssignedCostCenter',
        Name       => 'Assigned Cost Center',
        Searchable => 1,
        Input => {
            Type                          => 'CIClassReference',
            ReferencedCIClassName         => 'Cost Center',
            ReferencedCIClassLinkType     => 'RelevantTo',
            ReferencedCIClassLinkDirection => '',
            Required => 0,
        },
    },
    {
        Key        => 'CILocation',
        Name       => 'Asset Location',
```

```
        Searchable => 1,
        Input => {
            Type                           => 'CIClassReference',
            ReferencedCIClassName          => 'Location',
            ReferencedCIClassLinkType      => 'RelevantTo',
            ReferencedCIClassLinkDirection => '',
            Required => 0,
        },
    },
    {
        Key        => 'ResponsibleUser',
        Name       => 'Responsible user',
        Searchable => 1,
        Input => {
            Type      => 'Customer',
        },
    },
    {
        Key        => 'Maintenance',
        Name       => 'Maintenance',
        Searchable => 1,
        Input => {
            Type                           => 'CIClassReference',
            ReferencedCIClassName          => 'MaintenancePlan',
            ReferencedCIClassLinkType      => 'Includes',
            ReferencedCIClassLinkDirection => 'Reverse',
            Required                       => 1,
        },
        Sub => [
            {
                Key   => 'MaintenanceType',
                Name  => 'Maintenance type',
                Input => {
                    Type        => 'GeneralCatalog',
                    Class       => 'ITSM::ConfigItem::Maintenance::Type',
                    Translation => 1,
                    Required    => 0,
                },
            },
            {
                Key => 'ThresholdBased',
                Name => 'Threshold based',
                Input      => {
                    Type        => 'Dummy',
                    Required    => 0,
                },
                Sub => [
                    {
                        Key        => 'ThresholdAttribute',
                        Name       => 'Threshold attribute',
                        Input => {
                            Type      => 'Text',
                            Size      => 50,
```

```
                        MaxLength => 50,
                        Required  => 0,
                    },
                },
                {
                    Key        => 'NextCheckThreshold',
                    Name       => 'Next check threshold',
                    Input => {
                        Type      => 'Text',
                        Size      => 50,
                        MaxLength => 50,
                        Required  => 0,
                    },
                },
                {
                    Key        => 'NextCheckThresholdReminder',
                    Name       => 'Next check threshold reminder (%)',
                    Input => {
                        Type      => 'Text',
                        Size      => 50,
                        MaxLength => 50,
                        Required  => 0,
                    },
                },
                {
                    Key => 'NextCheckThresholdIncrease',
                    Name => 'Automatic threshold increase by',
                    Searchable => 1,
                    Input => {
                        Type => 'GeneralCatalog',
                        Class =>
'ITSM::ConfigItem::Maintenance::ThresholdIncrease',
                        Translation => 1,
                        Required => 0,
                    },
                    Sub => [
                        {
                            Key        => 'ManualThresholdIncrease',
                            Name       => 'Manual value for threshold increase',
                            Input => {
                                Type      => 'Text',
                                Size      => 50,
                                MaxLength => 50,
                                Required  => 0,
                                RegEx     => '^\d+$',
                            },
                        },
                    ],
                },
            ],
            CountMin => 0,
            CountDefault => 0,
            CountMax => 1,
```

```
    },
    {
        Key         => 'TicketTemplate',
        Name        => 'Ticket-Template',
        Searchable => 1,
        Input          => {
            Type       => 'TicketTemplateReference',
            Required   => 0,
        },
        CountMin     => 0,
        CountDefault => 0,
        CountMax     => 1,
    },
    {
        Key         => 'MaintenanceSupplier',
        Name        => 'Maintenance supplier',
        Input          => {
            Type       => 'CIACCustomerCompany',
            Required   => 0,
        },
        CountMin     => 0,
        CountMax     => 1,
    },
    {
        Key         => 'PlannedMaintenanceTime',
        Name        => 'Planned maintenance time (min)',
        Searchable => 1,
        Input          => {
            Type       => 'Text',
        },
        CountMin     => 0,
        CountMax     => 1,
        CountDefault => 0,
    },
    {
        Key => 'LastMaintenanceDate',
        Name => 'Last maintenance date',
        Searchable => 1,
        Input => {
            Type => 'Date',
            Required => 1,
        },
        CountMin => 0,
        CountMax => 1,
        CountDefault => 0,
    },
    {
        Key => 'MaintenanceNote',
        Name => 'Maintenance note',
        Searchable => 1,
        Input => {
            Type => 'TextArea',
            Required => 1,
```

```
                },
                CountMin => 0,
                CountMax => 1,
                CountDefault => 0,
            },
        ],
        CountMin => 0,
        CountMax => 10,
    },
];
```

If required, you can also combine both types of maintenance with each other within a CI class.

11.3.2 Composition of the Maintenance Intervals

In the following, you will find short examples of how the maintenance intervals are composed, and how exactly they are used.

Examples

'WeekdayPerMonth':

- Enter the following in the interval data: '[Every]:::[Weekdays]'
  - [Every] is a comma-separated list of the nth weekday in the month.
  - [Weekdays] is a comma-separated list of the days of the week.
  - Example showing every second Tuesday in the month: 2:::2

'Week', 'Month', 'Year':

- Here you only have to enter a number in the interval data.
  - Example showing 2nd month: 2

Please note: For 'Onetime' and 'Weekday', nothing is saved in the interval data.

12 MRO Bundle Add-On Module

12.1 Prelude

This manual is aimed at administrators who are going to work with KIX Professional MRO (Maintenance, Repair, Overhaul).

As a standard system, the software supports your entire maintenance and repair process for building services engineering, machines, systems, tools, and affected equipment.

Operational safety and processes become more structured. All processes required as part of regular certifications (ISO, TÜV) are facilitated.

Right after installation, the following basic functions that are often used in a service context are available to you, among others

- Device database/CMDB
- Job management
- Workflow management
- Maintenance planner
- Reporting

In addition to this manual, you can find further documents at: www.kixdesk.com[3]

3 https://www.kixdesk.com/en/

12.2 Installation

To be able to use KIX MRO, KIX must already be installed on your system. If this is not the case, you can find the applicable installation instructions at kixdesk.com[4].

12.2.1 Installation under Debian / Ubuntu

Please add the following line to file /etc/apt/sources.list:

```
deb http://packages.kixdesk.com/customers/xxxxxxxxxxx/debian kixpro2017 stable
```

Once you have integrated your personal repository and have met the requirement, install KIX MRO as follows:

```
apt-get update
apt-get install kixpro kixpro-bundle-mro
```

12.2.2 Installation under OpenSUSE / SLES

Add the repository of KIX Professional:

```
zypper addrepo http://packages.kixdesk.com/customers/xxxxxxxxxxx/suse/kixpro2017/
stable kixpro
```

Once you have integrated your personal repository and have met the requirement, install KIX MRO as follows:

```
zypper refresh

zypper install kixpro kixpro-bundle-mro
```

12.2.3 Installation under CentOS / Red Hat

Create a new repository file /etc/yum.repos.d/kixpro.repo with the following content:

```
[kixpro]

name=kixpro
```

4 http://www.kixdesk.com

```
baseurl=http://packages.kixdesk.com/customers/xxxxxxxxxxx/redhat/kixpro2017/stable

gpgcheck=1

enabled=1

gpgkey= proxy=_none_
```

Once you have integrated your personal repository and have met the requirement, install KIX MRO as follows:

```
yum check-updates

yum install kixpro kixpro-bundle-mro
```

12.2.4 Notes on Upgrade

Please note the following for all versions before KIX 17.7 which are upgraded to a higher version:

To transfer selected values from assigned objects to tickets, the package "DFITSMConfigItemReferenceFetchCIAttributes[5]" has been integrated. If the object provides several attribute values, only the first attribute value is taken over.

For the object classes "Computer", "Hardware", and "Network" there is now the option to assign a location to an object. These object classes have been expanded by the optional attribute "Location"

Note

- Object class definitions that are not present in the previous delivery state (if, for example, there is more than one version of the object class definition) are NOT changed by the update. The administrator must perform this manually themselves.
- Object class definitions that do not exist are not created by the update.

If you have adapted Sysconfig keys yourself, added translations independently, or created your own fields in the Ticketinfo sidebar, check these manually for the new wording. If errors occur in the interface, this is an indication that SysConfig keys may still have to be adapted. Here, check your ZZZAuto.pm in the config directory of your KIX for "old" terms. Here you can also find information for the SysConfig keys that have to be adapted.

Due to the new wording, the translation for ticket type "Incident" is delivered as "Wartung (reaktiv) (Maintenance (reactive))". +++ At certain points on the interface, the ticket type is not translated, however example: Sidebar). If you would like consistent wording here, too, you should adapt the name of the affected ticket type in the admin area. This impacts any ACLs and workflows that already exist, however. These then also have to subsequently be adapted at the points where this ticket type is used.

5 http://git.intra.cape-it.de/Customerprojects/Generic/DFITSMConfigItemReferenceFetchCIAttributes

Conclusion

All adaptations/additions/add-ons you have carried out yourself have to be checked and manually adjusted.

12.3 Components of KIX MRO Bundle

KIX MRO includes the modules Kanban Board, Maintenance Planner, and Field Service, along with certain specific MRO add-ons.

- Kanban Board (see page 195)
- Maintenance Planner (see page 195)
- Field Service (see page 195)
- KIX MRO Add-Ons (see page 196)
  - Generating CI Name (see page 196)
  - Event handler registration (see page 196)
  - Event handler configuration (see page 196)
  - Generating CI Number from Unique Attribute (see page 197)
  - Event handler registration (see page 197)
  - Event handler configuration (see page 197)

12.3.1 Kanban Board

The digital development of a sticky-note board: Visualize your work items, and create your own organization and project structure.

- Facilitates optimum deployment planning thanks to the fast and individually customizable assignment of jobs
- Permanent overview of availability and workload of your employees

12.3.2 Maintenance Planner

The purpose of maintenance work is to maximize the service life and planning of your technical equipment. Due to regular, event-based maintenance that can be planned, you minimize device and machine downtimes.

- Cyclical and threshold-based maintenance is supported
- It is possible to save templates (e.g. checklists) or maintenance contracts for each maintenance plan (paperless documentation)
- According to the maintenance plans, the system automatically creates the appropriate maintenance jobs for the equipment and the teams
- You are provided with an overview of all deadlines as the maintenance jobs are shown on a calendar

12.3.3 Field Service

Make the management, planning, and processing of jobs within technical field service easier.

Field Service optimizes all work steps from dispatching right up to the final documentation.

- Tour planning and management in a clear Kanban Board
- Map view of geographically distributed work orders and error reports
- End-to-end paperless documentation of the remediation, maintenance/repair orders
- Intuitive Field Service app for Android-based mobile devices including:
    - Offline mode with automatic synchronization,
    - Checklist function
    - Digital signature and
    - Multiple encryption

You can find further information and configuration options at docs.kixdesk.com[6].

12.3.4 KIX MRO Add-Ons

Once you have installed the MRO bundle, you can adapt the basic parameters of this add-on package as required.

To configure the add-on, open "SysConfig" in the Admin area. Select SysConfig group "MROExtensions". A default configuration is already contained in the package. You can change this as required. The relevant SysConfig options are listed in the following sections.

12.3.4.1 Generating CI Name

The automatic creation of a CI name is activated for all new CIs of the class "CostCenter" by default.

12.3.4.2 Event handler registration

SysConfig: ITSMConfigItem :: EventModulePre # ## 200-GenerateNameFromAttributes

- Registration of CI event handler.

> **Please note**
>
> If you do not want to automatically create the name of a new CI, deactivate this registration option.

12.3.4.3 Event handler configuration

SysConfig: MRO ::GenerateNameFromAttributes###...

- These configuration options are required if you use the event handler. The descriptions outline what you have to do and how you configure it.

6 http://docs.kixdesk.com

12.3.4.4 Generating CI Number from Unique Attribute

By default, the automatic creation of a unique CI number on the basis of a unique attribute value of a new CI is activated for all new CIs of the class "CostCenter".

12.3.4.5 Event handler registration

SysConfig: ITSMConfigItem :: EventModulePost ### 100-SetCINumberByUniqueAttribute

- Registration of CI event handler.

> **Please note**
>
> If you do not want to automatically create the CI number from an attribute of a new CI, deactivate this registration.

12.3.4.6 Event handler configuration

SysConfig: MRO ::SetCINumberByUniqueAttribute :: ...

- These configuration options are required if the event handler is used. The descriptions outline what you have to do and how you must configure.

13 Security Management

13.1 General Information on Security Management

Security Management makes it possible to map the CMDB of KIX (as leading system) in external Security Management systems (e.g. verinice.PRO) as far as possible. The security officer using the external system can then, for example, apply the IT-Grundschutz (IT Baseline Protection) to the transferred CIs.

Using the REST interface of verinice.PRO and KIX, the measures to be implemented that are necessary to comply with the Baseline Protection are subsequently recorded as individual tickets in KIX.

The IT department, which also uses KIX, can then process these tickets accordingly and implement the associated measures, as well as log them on the ticket.

The process of collaboration between the two departments in the company is thereby supported and a seamless and structured handover and documentation of tasks is ensured.

The module supports the following three aspects in verinice.PRO through the backends that are already integrated. It is possible to use several backends at the same time.

Aspect	Responsible backend
German Federal Office for Information Security Baseline Protection (BSI Grundschutz)	BSI
Information Security Management	ISM
Modernized BSI Baseline Protection	BSIModern

Please note

The relevant text data is already provided to display the description texts for the measures from verinice.PRO ("BSI-Grundschutz" aspect) in the article content. This comes from the "15. Ergänzungslieferung" of the IT-Grundschutz Katalog (15th Version of the IT Baseline Protection Catalog), which should also be used in verinice.PRO (the catalog can be downloaded for free at "verinice.com[7]").

If there is a more up-to-date version (not the "IT-Grundschutz-Kompendium" (IT Baseline Protection Compendium) – this is for modernized BSI Baseline Protection), the new texts can be used by unpacking the updated catalog (zip) and saving the "m" folder contained therein in "kix-home/SecurityManagement/var/verinice_BSI".

7 https://verinice.com/

13.2 Installation of Security Management Add-On Module

13.2.1 Prerequisites

- KIX Professional 17.x – same major and minor version as SecurityManagement. If your Security Management has version 17.11.1, KIX Professional 17.11.0 at the least must be installed.
- verinice.PRO 1.2x with applicable REST API is available

13.2.2 Package Installation

Installation on Debian / Ubuntu

Please add the following line to file /etc/apt/sources.list:

deb http://packages.kixdesk.com/customers/xxxxxxxxxxx/debian kixpro2017 stable

Once you have integrated your personal repository and have met the requirement, install Security Management as follows:

apt-get update apt-get install kixpro-security-management

Installation under OpenSUSE / SLES

Add the repository of KIX Professional:

zypper addrepo http://packages.kixdesk.com/customers/xxxxxxxxxxx/suse/kixpro2017/stable[8] kixpro

Once you have integrated your personal repository and have met the requirement, install

Security Management as follows:

zypper refresh zypper install kixpro-security-management

Installation on CentOS / Red Hat

Create a new repository file /etc/yum.repos.d/kixpro.repo with the following content:

[kixpro] name=kixpro baseurl=http://packages.kixdesk.com/customers/xxxxxxxxxxx/redhat/kixpro2017/stable[9] gpgcheck=1

Once you have integrated your personal repository and have met the requirement, install Security Management as follows:

yum check-updates yum install kixpro-security-management

8 http://packages.kixdesk.com/customers/xxxxxxxxxxx/suse/kixpro2016/stable
9 http://packages.kixdesk.com/customers/xxxxxxxxxxx/redhat/kixpro2016/stable

13.2.3 Migration of Version 17.4.x

SecurityManagement 17.4.x was operated under the old name "verinice4KIX". From version 17.5.x, it is now operating under "Security Management". When updating verinice4KIX, the following adjustments have to be made so that everything works seamlessly together:

13.2.3.1 Changes to Relevant SysConfig Settings

- Core::VeriniceInterface###RESTServiceURL
  - Reset to default, as key has been renamed. The following connection settings may also have to be checked and modified.
- Core::VeriniceInterface###SOAPServiceURL
  - Reset to default, as key has been renamed. The following connection settings may also have to be checked and modified.

The following keys have to be reset due to regular changes in the configuration of dynamic fields:

- Ticket::Frontend::KIXSidebarVeriniceTaskMainData###DynamicField
- Ticket::Frontend::AgentTicketZoomTabVeriniceTaskData###DynamicField
- Daemon::SchedulerCronTaskManager::Task###CheckForVeriniceTask
  - Reset to default, as key has been renamed. The interval may have to be modified here.

13.2.3.2 Modifications in Admin Area

New dynamic fields are created due to changes in backend structure:

New field	Content
TaskChapter	Contains the measures title from verinice.PRO for the task. Depending on the backend used, not all fields are relevant or have a counterpart in verinice.PRO.
TaskConfigItemID	Contains IDs of CIs involved from KIX.
TaskDescription	Contains the measures chapter from verinice.PRO for the task.
TaskExternalCIName	Contains CI names of external system.
TaskID	Contains the ID for the task of the external system (e.g. "dbid" of a measure from verinice.PRO).

New field	Content
TaskImplementationByDate	Contains the date of the measure from verinice.PRO for task implementation.
TaskImplementationExplanation	Contains the explanation of the measure from verinice.PRO for task implementation.
TaskImplementationState	Contains the state of the measure from verinice.PRO for task implementation. Depending on the backend used, not all fields are relevant or have a counterpart in verinice.PRO.
TaskTitle	Contains the measures title from verinice.PRO for the task. Depending on the backend used, not all fields are relevant or have a counterpart in verinice.PRO.

Note

The mapping for KIX attributes to verinice attributes should always be checked.
Only update relevant – old dynamic fields can thereby be set as **invalid**:

- TaskVeriniceCIName
- TaskMeasureTitle
- TaskMeasureChapter
- TaskMeasureDescription
- TaskMeasureImplementationImplementation
- TaskMeasureImplementationByDate
- TaskMeasureImplementationExplanation
- TaskMeasureVeriniceID

13.3 Administration of Security Management

You can configure settings for using your Security Management in the Admin area of KIX via SysConfig. The individual keys are explained in more detail in the following chapters.

13.3.1 Calling up Tasks in KIX

If you would like to manually synchronize modified objects from verinice.PRO to KIX, in a console, execute the following command:

```
bin/kix.Console.pl Maint::Ticket::CheckForTasks
```

This command will otherwise be executed by the KIX daemon every 5 minutes.

13.3.2 New Objects

13.3.2.1 New Access Control List (ACL)

During installation, a new ACL is entered in the Admin area as a configuration. This hides the new ticket tab "verinice Aufgabeninformationen (verinice task information)" if a ticket is not a verinice.PRO task ticket.

13.3.2.2 New Ticket Type

During installation, a new ticket type ("veriniceTask") is created. This is used by default to create new verinice.PRO task tickets.

13.3.2.3 New Dynamic Fields

New dynamic fields are created during installation:

Name	**Function**
TaskConfigItemID	Contains IDs of CIs involved from KIX.
TaskExternalCIName	Contains CI names of external system.
TaskID	Contains the ID for the task of the external system (e.g. "dbid" of a measure from verinice.PRO).
Task...	Contains general values for the task (e.g.: the measures title from verinice.PRO). Depending on the backend used (see Chapter), not all fields are relevant or have a counterpart in verinice.PRO.
TaskImplementation...	Contains values for task implementation (e.g.: implementation date of the measure from verinice.PRO). Depending on the backend used (see Chapter), not all fields are relevant or have a counterpart in verinice.PRO.

"TaskConfigItemID", "TaskExternalCIName", and "TaskID" are set automatically as far as possible. All other fields are populated through configured assignments.

13.3.2.4 New CI Classes

During installation, new CI classes are created:

Name	Function
Server	Serves to represent a Configuration Item in verinice (e.g. of the server category of the BSI-Grundschutz aspect).
ClientGroup	Serves to represent the client category of the BSI-Grundschutz aspect of verinice.PRO. A "Client" in verinice.PRO is a group of CIs with identical properties.

13.3.2.5 New Ticket States

During installation, six new ticket states are created. A simple state workflow is also available.

Name	Function
Task new	Is set by default for new task tickets.
Task assigned	Can be used to show that the task ticket has been assigned to an owner.
Task implementation	Can be used to show that the task ticket is being implemented.
Task rejected	Can be used to show that the task ticket is not being implemented or that it is no longer relevant or that the incorrect owner was selected.
Task done	Can be used to show that the task ticket has been completed and the measure implemented.
Task implementation accepted	Can be used to show that the measure of the task ticket has been implemented satisfactorily and has therefore been accepted.

13.3.3 Configuration of Security Management

- SysConfig Options (see page 207)
- SOAP Interface – URL (see page 207)
- SOAP Interface – User Name (see page 207)
- SOAP Interface – Password (see page 207)
- SOAP Interface – SSL (see page 208)
- REST Interface – URL (see page 208)
- REST Interface – User Name (see page 208)
- REST Interface – Password (see page 208)
- REST Interface – SSL (see page 208)
- Source Name (see page 209)
- verinice Backends (see page 209)
- Synchronization – Relevant CI Classes (see page 209)
- Synchronization – Attribute Mapping (see page 209)
- Synchronization – Attribute Value Mapping (see page 210)
- Synchronization – Link Type Mapping (see page 210)
- Event Module for Ticket Changes (see page 210)
- Ticket Type (see page 210)
- Ticket State Workflow (see page 211)
- Task Ticket – BSI Backend – Core Data (see page 211)
- Task Ticket – BSI Backend – Article (see page 211)
- Task Ticket – BSI Backend – Dynamic Fields – Mapping (see page 211)
- Task Ticket – BSI Backend – Dynamic Fields – For Ticket Change (see page 212)
- Task Ticket – BSIModern Backend – Core Data (see page 212)
- Task Ticket – BSIModern Backend – Article (see page 212)
- Task Ticket – BSIModern Backend – Dynamic Fields – Mapping (see page 213)
- Task Ticket – BSIModern Backend – Dynamic Fields – For Ticket Change (see page 213)
- Mapping KIX Identifier / verinice Identifier (see page 213)
- Task Ticket – ISM Backend – Core Data (see page 213)
- Task Ticket – ISM Backend – Article (see page 214)
- Task Ticket – ISM Backend – Dynamic Fields – Mapping (see page 214)
- Task Ticket – ISM Backend – Dynamic Fields – For Ticket Change (see page 214)
- Task Ticket – Creation by Scheduler Module (see page 214)
- Sidebar – Dynamic Fields (see page 215)
- Ticket Tab – Dynamic Fields (see page 215)
- Ticket Tab – Further Settings (see page 215)

13.3.3.1 SysConfig Options

Following installation, you can configure certain settings in SysConfig.

For this, open SysConfig in the Admin area of KIX. Select the "Security Management" main group and one of the following subgroups:

- Core::Ticket
- Core::VeriniceInterface
- Daemon::SchedulerCronTaskManager::Task
- Frontend::Agent::KIXSidebarBackend
- Frontend::Agent::Ticket::ViewTicketZoomTabVeriniceTaskData
- VeriniceSync::BSI
- VeriniceSync::BSIModern
- VeriniceSync::ISM
- VeriniceTask::BSI
- VeriniceTask::BSIModern
- VeriniceTask::ISM

Default configuration is already provided. You can adapt this in accordance with your requirements. The following chapters explain the SysConfig options that can be configured.

> SysConfig option "Ticket::EventModulePost###ITSMConfigItemLinkAdd" is activated by installation.

13.3.3.2 SOAP Interface – URL

SysConfig option: Core::VeriniceInterface###SOAPServiceURL

Defines the service URL for SOAP requests for CI synchronization with verinice.PRO.

> This option is required and must be adapted.

13.3.3.3 SOAP Interface – User Name

SysConfig option: Core::VeriniceInterface###SOAPAuthenticationName

Defines the user name for accessing the SOAP interface of verinice.PRO.

> This option is necessary and default value is "admin".

13.3.3.4 SOAP Interface – Password

SysConfig option: Core::VeriniceInterface###SOAPAuthenticationPassword

Defines the password for accessing the SOAP interface of verinice.PRO.

> This option is necessary and default value is "secret".

13.3.3.5 SOAP Interface – SSL

SysConfig option: Core::VeriniceInterface###SOAPNoSSLVerify

Deactivates SSL certificate validation for the SOAP interface, e.g. if a transparent HTTPS proxy is used.

> Activate this option at your own risk!

13.3.3.6 REST Interface – URL

SysConfig option: Core::VeriniceInterface###RESTServiceURL

Defines the service URL (Host) for the REST API of verinice (with port "8081" by default).

> This option is required and must be adapted.

13.3.3.7 REST Interface – User Name

SysConfig option: Core::VeriniceInterface###RESTAuthenticationName

Defines the user name for accessing the REST API of verinice.PRO.

> This option is necessary and default value is "verinice".

13.3.3.8 REST Interface – Password

SysConfig option: Core::VeriniceInterface###RESTAuthenticationPassword

Defines the password for accessing the REST API of verinice.PRO.

> This option is necessary and default value is "verinice".

13.3.3.9 REST Interface – SSL

SysConfig option: Core::VeriniceInterface###RESTNoSSLVerify

Deactivates SSL certificate validation for the REST API, e.g. if a transparent HTTPS proxy is used.

Activate this option at your own risk!

13.3.3.10 Source Name

SysConfig option: Core::VeriniceInterface###SourceName

Defines the name of the source. The name is used for synchronization of the KIX CMDB with verinice.PRO. This means CIs that were created with the same source name through a previous synchronization can be determined and updated in verinice.PRO. Furthermore, the name serves to link verinice.PRO task tickets to synchronized CIs in KIX.

This option is necessary and default value is "KIX".

13.3.3.11 verinice Backends

SysConfig option: Core::ActiveBackends

Defines which backends are active for verinice. Only active backends are taken into account for synchronization, task ticket creation, and the feedback of ticket changes to verinice.PRO.

This option is required.

13.3.3.12 Synchronization – Relevant CI Classes

SysConfig option: VeriniceSync::[Backend_Name]###RelevantCIClasses

Defines which CI classes can be used in the synchronization view. The key must be the CI class and the content the ID of the corresponding object in verinice (e.g. "server", "asset", or "it-system").

This option is required. The values which are possible from verinice.PRO can be found in the file "SNCA.xml" from verinice.PRO.

13.3.3.13 Synchronization – Attribute Mapping

SysConfig option: VeriniceSync::[Backend_Name]###AttributeMapping

Defines the mapping for the synchronization of CI attributes with the applicable object attributes in verinice.PRO. The key must be the CI class, followed by three colons and the CI attribute. The content should be the ID of the applicable object attribute in verinice.PRO.

This option is required. The values which are possible from verinice.PRO can be found in the file "SNCA.xml" from verinice.PRO.

13.3.3.14 Synchronization – Attribute Value Mapping

SysConfig option: VeriniceSync::[Backend_Name]###AttributeOptionMapping

Defines the mapping for synchronization of the available options of a CI attribute (dropdown) with the options of the applicable object attribute in verinice.PRO. The key must be the CI class followed by three colons and the CI attribute, followed by a further three colons and the option itself. The content should be the ID of the applicable object attribute option in verinice.PRO.

> The values which are possible from verinice can be found in the file "SNCA.xml" from verinice.PRO.

13.3.3.15 Synchronization – Link Type Mapping

SysConfig option: VeriniceSync::<BackendName>###LinkTypeMapping

These options determine how the mapping of the link types is to look from KIX to verinice.PRO. The keys have the form <Source-CI-Class>::<Target-CI-Class>::<Link-Type>. Fixed values from verinice.PRO are the content (see SNCA.xml there)

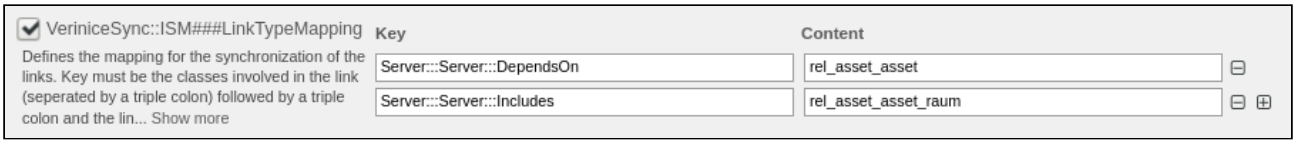


Figure: Link Type Mapping

13.3.3.16 Event Module for Ticket Changes

SysConfig option: Ticket::EventModulePost###444-VeriniceTaskTicketForwarding

Event module to transfer changes to a task ticket to verinice.PRO. Under "SetKIXWorkflowOfTaskOnTicketState" you can define which ticket state sets which KIX workflow state in verinice.PRO.

> By default, when the task is assigned ("Task assigned"), the KIX workflow state in verinice.PRO is set to "Ongoing" and when the task implementation is accepted ("Task implementation accepted"), it is set to "Ended".

13.3.3.17 Ticket Type

SysConfig option: VeriniceTicketType

Ticket type for verinice task tickets.

This option is necessary and default value is "veriniceTask".

13.3.3.18 Ticket State Workflow

SysConfig option: TicketStateWorkflowExtension###veriniceTask

Defines the ticket state workflow for verinice task tickets.

So that the workflow is applied and the ticket states restricted applicably, SysConfig option "Ticket::Acl::Module###950_TicketStateWorkflow" must be activated.

13.3.3.19 Task Ticket – BSI Backend – Core Data

SysConfig option: VeriniceTask::BSI###TicketData

Defines the core data of a new verinice.PRO task ticket. For "Titel (title)" placeholders such as "<VERINICE_mnums_id>" or "<VERINICE_server_name>" are possible for verinice attributes. The possible attribute placeholders for measures ("mnums"), modules ("bstumsetzung"), and CIs ("server" or "client") can be found in the file "SNCA.xml" from verinice.PRO.

This option is required.

13.3.3.20 Task Ticket – BSI Backend – Article

SysConfig option: VeriniceTask::BSI###ArticleData and VeriniceTask::BSI###ArticleDataBody

Article data of a newly created verinice task ticket. For "Subject" or the body text, placeholders such as "<VERINICE_mnums_id>" or "<VERINICE_server_name>" are possible for verinice.PRO attributes. The possible attribute placeholders for measures ("mnums"), modules ("bstumsetzung"), and CIs ("server" or "client") can be found in the file "SNCA.xml" from verinice.PRO.

This option is required. For the body text, the special placeholder "<VERINICE_mnums_bsi_description>" is also possible (already entered by default). The description text of the measures from the BSI Baseline Protection Catalog (see Chapter) is thereby entered at the applicable point.

13.3.3.21 Task Ticket – BSI Backend – Dynamic Fields – Mapping

SysConfig option: VeriniceTask::BSI###TicketDataDynamicFieldMapping

Defines the mapping for dynamic fields of new verinice.PRO task tickets. The key is the dynamic field and the content the verinice.PRO attribute. Possible verinice.PRO attributes can be found in the file "SNCA.xml" from verinice.PRO.

This option is required.

13.3.3.22 Task Ticket – BSI Backend – Dynamic Fields – For Ticket Change

SysConfig option: VeriniceTask::BSI###DynamicFieldsForResponse

Defines which dynamic fields are transferred to verinice.PRO by the event module (see Chapter) in the event of changes to task tickets (1 = active, 0 = inactive). Mapping (see Chapter) must be configured for the dynamic fields.

This option is required.

13.3.3.23 Task Ticket – BSIModern Backend – Core Data

SysConfig option: VeriniceTask::BSIModern###TicketData

Defines the core data of a new verinice task ticket. For "Titel (title)" placeholders such as "<VERINICE_bp_safeguard_id>" or "<VERINICE_bp_itsystem_name>" are possible for verinice attributes. To define values for a certain verinice object type only, the KIX identifier (followed by three colons) can be placed in front of the ticket attribute (e.g. "Module:::Title"). Entries without KIX identifier are the default.

This option is required.

13.3.3.24 Task Ticket – BSIModern Backend – Article

SysConfig option: VeriniceTask::BSIModern###ArticleData

Article data of a newly created verinice.PRO task ticket. For "Subject" and "Body" placeholders such as "<VERINICE_bp_safeguard_id>" or "<VERINICE_bp_itsystem_name>" are possible for verinice.PRO attributes. For "Body" certain DOM elements are also possible (e.g.
, for "line breaks" – if the "ContentType" is specified as "text/html"). To define values for a certain verinice.PRO object type only, the KIX identifier (followed by three colons) can be placed in front of the article attribute (e.g. "Module:::Subject"). Entries without KIX identifier are the default.

This option is required.

13.3.3.25 Task Ticket – BSIModern Backend – Dynamic Fields – Mapping

SysConfig option: VeriniceTask::BSIModern###TicketDataDynamicFieldMapping

Defines the mapping for dynamic fields of new verinice.PRO task tickets. The key is the dynamic field and the content the verinice.PRO attribute. To define values for a certain verinice.PRO object type only, the KIX identifier (followed by three colons) can be placed in front of the dynamic field (e.g. "Module:::TaskTitle"). Entries without KIX identifier are the default.

This option is required.

13.3.3.26 Task Ticket – BSIModern Backend – Dynamic Fields – For Ticket Change

SysConfig option: VeriniceTask::BSIModern###DynamicFieldsForResponse

Defines which dynamic fields are transferred to verinice.PRO by the event module (see Chapter) in the event of changes to task tickets (1 = active, 0 = inactive). Mapping (see Chapter) must be configured for the dynamic fields.

This option is required.

13.3.3.27 Mapping KIX Identifier / verinice Identifier

SysConfig option: VeriniceTask::BSIModern###VeriniceObjects

This option defines which objects from verinice.PRO are considered for verinice.PRO task tickets in KIX. The "Key" is the KIX-internal identifier and "Value" the applicable object identifier (type) in verinice.PRO.

13.3.3.28 Task Ticket – ISM Backend – Core Data

SysConfig option: VeriniceTask::ISM###TicketData

Defines the core data of a new verinice.PRO task ticket. For "Titel (title)" placeholders such as "<VERINICE_control_name>" or "<VERINICE_asset_name>" are possible for verinice.PRO attributes. The possible attribute placeholders for controls ("control"), control group ("controlgroup"), and CIs ("asset") can be found in the file "SNCA.xml" from verinice.PRO.

This option is required.

13.3.3.29 Task Ticket – ISM Backend – Article

SysConfig option: VeriniceTask::ISM###ArticleData and VeriniceTask::ISM###ArticleDataBody

Article data of a newly created verinice task ticket. For "Subject" or the body text, placeholders such as "<VERINICE_control_name>" or "<VERINICE_asset_name>" are possible for verinice attributes. The possible attribute placeholders for controls ("control"), control groups ("controlgroup"), and CIs ("asset") can be found in the file "SNCA.xml" from verinice.PRO.

This option is required.

13.3.3.30 Task Ticket – ISM Backend – Dynamic Fields – Mapping

SysConfig option: VeriniceTask::ISM###TicketDataDynamicFieldMapping

Defines the mapping for dynamic fields of new verinice.PRO task tickets. The key is the dynamic field and the content the verinice.PRO attribute. Possible verinice.PRO attributes can be found in the file "SNCA.xml" from verinice.PRO.

This option is required.

13.3.3.31 Task Ticket – ISM Backend – Dynamic Fields – For Ticket Change

SysConfig option: VeriniceTask::ISM###DynamicFieldsForResponse

Defines which dynamic fields are transferred to verinice.PRO by the event module (see Chapter) in the event of changes to task tickets (1 = active, 0 = inactive). Mapping (see Chapter) must be configured for the dynamic fields.

This option is required.

13.3.3.32 Task Ticket – Creation by Scheduler Module

SysConfig option: Daemon::SchedulerCronTaskManager::Task###CheckForVeriniceTask

Scheduler module registration (Scheduler) to check whether there are new verinice.PRO tasks and create new tickets for this as applicable. The content for "Schedule" is a cron notation.

The check is performed every 5 minutes by default.

13.3.3.33 Sidebar – Dynamic Fields

SysConfig option: Ticket::Frontend::KIXSidebarVeriniceTaskMainData###DynamicField

Determines the dynamic fields that are shown in the "verinice Aufgaben-hauptinformationen (verinice task main information)" sidebar in the agent frontend. Possible settings are "0" for "nicht anzeigen (do not show)" and "1" for "anzeigen (show)".

13.3.3.34 Ticket Tab – Dynamic Fields

SysConfig option: Ticket::Frontend::AgentTicketZoomTabVeriniceTaskData###DynamicField

Determines the dynamic fields that are shown in the "verinice Aufgabeninformationen (verinice task information)" tab in the agent frontend. Possible settings are "0" for "nicht anzeigen (do not show)" and "1" for "anzeigen (show)", as well as "2" for "anzeigen und als Pflichtfeld markieren (show and mark as mandatory field)".

13.3.3.35 Ticket Tab – Further Settings

SysConfig option: Ticket::Frontend::AgentTicketZoomTabVeriniceTaskData###***

These options determine which rights a user must have to be able to use the tab, whether the ticket must be locked, whether the ticket state can be changed, and which ticket states are available in the process (based on their state types).

> ⓘ Some options are required.

13.3.4 Preparation for Use

13.3.4.1 Preparation for verinice

To synchronize CIs and determine verinice tasks, the respective interface URLs (see Chapter and), and the users and passwords for this if applicable (see Chapter and as well as and), have to be adapted to their environment in SysConfig.

Objects of the category "Client" in the BSI-Grundschutz aspect in verinice represent groups of CIs with the same configuration. The new class "ClientGroup" has been created to depict this in KIX. So that this can determine its corresponding "Clients" a new attribute has to be inserted into the class definition in the applicable CI classes (for example "Computer"):

```
{
Key => 'ClientGroup',
Name => 'ClientGroup',
Searchable => 1,
Input => {
    Type => 'CIClassReference',
    ReferencedCIClassName => 'ClientGroup',
    ReferencedCIClassLinkType => 'Includes',
    ReferencedCIClassLinkDirection => 'Reverse',
    ReferencedCIClassReferenceAttributeKey => 'Name',
    Required => 1,
    },
},
```

In the "Client" CIs concerned in KIX, the relevant "ClientGroup" CI in the new attribute must then be selected. This means the CIs are also linked with each other. In verinice.PRO, after synchronizing the "ClientGroup" CI, the number of "Clients" (the linked CIs of the "ClientGroup" CI) is visible as an attribute.

13.3.4.2 Synchronization

So that CIs can be synchronized, 3 prerequisites must be met. They must belong to a class configured in SysConfig (see Chapter), mapping for this class should be available (see Chapter and), and the attribute "Synchronization with verinice" must be set to "Yes" for the applicable Configuration Item:

```
{
Key => 'VeriniceSync',
Name => 'Synchronization with verinice',
Input => {
    Type => 'GeneralCatalog',
    Class => 'ITSM::ConfigItem::YesNo',
    Translation => 1,
```

```
    Required => 1,
    },
},
```

Due to the installation of the "Security Management" module, the "Server" class (copy of the default class "Computer") has been created. This already contains certain verinice.PRO-related attributes (e.g. protection requirements), as well as the attribute "Synchronization with verinice". The "Server" class is also pre-configured for all 3 backends provided (please see Chapter).

In the "CMDB" menu, the new sub-item "verinice Sync" is available, which can be used to open the synchronization overview. This displays all the configured classes, the backends that they are configured for, and the number of CIs to be synchronized. Following synchronization, a window appears with a result showing the number of Configuration Items that have been created or changed in verinice.PRO. Any errors are also shown with a reference to the respective relevant error log.

Synchronization can also be performed manually via a terminal with the CLI module "Maint::ITSM::SyncConfigItemsToVerinice".

In verinice, the new Configuration Items in the respective aspect then only have to be moved from the "imported objects" into the applicable relevant object category using "Cut" and "Paste".

Please note

If a class is configured for several backends, the relevant CIs are created/updated for each backend concerned. This means a Configuration Item in KIX can be present up to 3x (in different forms depending on the backend and mapping) in verinice.PRO.

13.3.4.3 verinice Task Tickets

For tickets to be created, in verinice.PRO, "modules" from the "IT Baseline Protection Catalog" ("BSI-Grundschutz") or the "IT Baseline Protection Compendium" ("Modernisierter BSI-Grundschutz") or controls (ISM aspect) must be assigned to the synchronized Configuration Items. For the measures/controls contained in the modules, the attribute "KIX Workflow Status (KIX Workflow State)" must be set to "Starte Workflow (Start workflow)".

In KIX, the scheduler queries by default every 5 minutes using REST whether there are "Tasks" (measures/controls) in verinice.PRO with a started KIX Workflow. If there are applicable "Tasks", a new verinice task ticket is created in KIX for each "Task". If there is already an open ticket for this "Task", only the dynamic fields are updated.

The CLI module "Maint::Ticket::CheckForTasks" used in the scheduler can also be executed manually via a terminal.

When creating a ticket, the next free "KIX Ticket X" attribute of the measure/control in verinice is populated with the link to the ticket and the ticket number. This means the link to the KIX tickets is also saved in verinice.

As long as the workflow state remains on "Starte Workflow (Start Workflow)", during the next request (via the scheduler or also manually via terminal), the measure/control will be considered again and, if an open ticket for
this "Task" no longer exists in KIX, another verinice.PRO task ticket is created for this "Task" and the next free "KIX Ticket X" attribute populated in verinice.PRO.

Once a task has been implemented, the implementation state and further information can be transferred to verinice.PRO by setting the relevant ticket state for the verinice task ticket in KIX.

14 Telephone Integration

14.1 Configuration of Telephone Integration Add-On Module

Once you have installed the package, you can configure all the necessary parameters for communicating with the telephone system in SysConfig. Select SysConfig group "CTI4KIX". After page has reloaded, select subgroup "Core".

During installation, two new permission groups are created: "CTI" and "CTIManager" to manage access to the CTI functions. The group "CTI" is reserved for future use. The "CTIManager" module is a log of all current or past CTI activities. Only agents with "rw" permissions in the group "CTIManager" have access to this.

Following installation, three new widgets "TelefonID (PhoneID)" /"PhoneIDType" / "PhoneIP" are available in an agent's personal settings. Adapt these settings in line with your circumstances.

- Selection of Backend (see page 220)
- Customer User PhoneID Type Mapping (see page 220)
- Customer User PhoneID Type Mapping - Standard (see page 221)
- Customer User PhoneID Mapping (see page 221)
- Agent PhoneID Type (see page 221)
- Incoming Calls – Filter (see page 221)
- Recognition of Extension Number (see page 222)
- Completion of Number Extension (see page 222)
- Default Country Code (see page 222)
- Settings for Outgoing Calls (see page 222)
- Displaying Telephone Number Selection (see page 223)
- Search Type for Searching for Agents and Contact Persons for Incoming Calls (see page 223)
- Searching for Customers (see page 223)

14.1.1 Selection of Backend

- SysConfig setting: CTI::BackendModule

Using this setting, you define which backend is to be used for CTI. The HTTP backend (Action URL handling) is available and selected by default. If you have installed other backend modules in KIX Professional, these can also be selected.

14.1.2 Customer User PhoneID Type Mapping

- SysConfig setting: CTI::Parameters###CustomerUserPhoneIDTypeAttribute

Using this setting, you define which attribute of a customer user specifies the type of telephone number. The type can be a telephone number, a Session Initiation Protocol (SIP), or a MAC address, depending on your installed backend.

Default value is "UserPhoneType". A different type can be used for each customer.

14.1.3 Customer User PhoneID Type Mapping - Standard

- SysConfig setting: CTI::Parameters###CustomerUserPhoneIDTypeDefault

Using this setting, you define the default type of the telephone number if no type is specified in "CTI::Parameters###CustomerUserPhoneIDTypeAttribute".

14.1.4 Customer User PhoneID Mapping

- SysConfig setting: CTI::Parameters###CustomerUserPhoneIDAttributes
- SysConfig setting: CTI::Parameters###CustomerUserPhoneIDAttributeDefault

Using this setting, you define which attributes of a customer user can be used as the telephone number. The type can be a telephone number, a Session Initiation Protocol (SIP), or a MAC address, depending on your installed backend.

"UserPhone" and "UserMobile" are used by default.

14.1.5 Agent PhoneID Type

- SysConfig setting: CTI::Parameters###AgentPhoneIDTypeDefault

Using this setting, you define the type of the telephone number of an agent. The type can be a telephone number, a Session Initiation Protocol (SIP), or a MAC address, depending on your installed backend.

Default value is "Telephone number".

14.1.6 Incoming Calls – Filter

- SysConfig setting: CTI::Parameters###IncomingPhoneIDSrcFilterRegEx
- SysConfig setting: CTI::Parameters###IncomingPhoneIDDestFilterRegEx

For incoming calls, parts of the source or target parameters can be restricted by a filter. If the format of the parameters does not match the value in the database, this can be adapted by RegEx.

An example of the source:

- Value in the database: 00493715347
- Format of the incoming call: #000134@00493715347

Here, the RegEx must extract the part following "@" so that the value matches the value in the database.

14.1.7 Recognition of Extension Number

- SysConfig setting: CTI::Parameters###ExtensionNumberLengthThreshold

Using this setting, you define the length threshold of a phone number. If the number is within this threshold, it is interpreted as an extension number.

Default value is "3".

14.1.8 Completion of Number Extension

- SysConfig setting: CTI::Parameters###ExtensionNumberCompletion

Using this setting, you define the first part of a telephone number to complete an extension number.

14.1.9 Default Country Code

- SysConfig setting: CTI::Parameters###DefaultCountryCode

Using this setting, you define the default country code.

14.1.10 Settings for Outgoing Calls

The following settings are required to configure the functionality for outgoing calls. Please note that outgoing calls are only possible when a telephone number is available for the contact person of the ticket AND for the current agent. If this is not the case, the applicable dialog is not displayed.

Please note that all settings with "HTTP" only apply for the HTTP backend. If you use another backend, please refer to the documentation for this backend.

- SysConfig setting: CTI::Backend::HTTP::Parameters###DialFrom

Depending on your infrastructure, the server can take over the central communication for dialing. It is possible that the server is sometimes not able to dial because of a firewall, for example. This setting can then be changed to "Client". This means a user can directly dial from the web browser, initiated via Javascript.

- SysConfig setting:CTI::Backend::HTTP::Parameters###DialOutURL

This setting defines the URL template that is used for outgoing calls. If the telephone system supports "dial-byURL", KIX Professional can automatically archive the outgoing calls of agents, and assign them to the applicable tasks. This URL contains the telephone number of the agent and destination phone number.

- SysConfig setting: CTI::Backend::HTTP::Parameters###DialOutProxyURL

Using this setting, you define the proxy URL for outgoing calls. This setting is only useful if the value "Server" has been selected in "CTI::Backend::HTTP::Parameters###DialFrom". The setting must be deactivated if no proxy is used.

- SysConfig setting: CTI::Backend::HTTP::Parameters###ClientDialServerFeedbackURL

Using this setting, you define the URL template that is used for client-based dialing in order to inform the server.

14.1.11 Displaying Telephone Number Selection

- SysConfig setting: CTI::Backend::HTTP::Parameters###AgentPhoneOutboundEnablePhoneIDSelection

A selection list can be displayed for an outgoing call. In the list, the agent can select the number from which the outgoing call is initiated.

14.1.12 Search Type for Searching for Agents and Contact Persons for Incoming Calls

- SysConfig setting: CTI::Parameters###SearchType

Using this setting, you define the search type that is to be used to search for agents or contact persons for incoming calls. "Live" is slower but performs a live search and updates the metadata in the database. "Meta" is faster but only uses the metadata saved in the database.

14.1.13 Searching for Customers

- SysConfig setting: CTI:Parameters###AgentPhoneNewCTIActionForCustomerUserLookUp

Using this setting, you define which CTI action triggers the search for customers when an agent opens a new ticket during an incoming call. The action "CONNECTED" is set by default, which means that an incoming call is connected to a user's telephone.

14.2 Installation of Telephone Integration Add-On Module

14.2.1 Requirements for the Telephone System

14.2.1.1 Basic Requirements

In order to be able to use the CTI add-on for KIX, the telephone system must fundamentally be able to address a particular URL (action URL) for certain actions (at least INCOMING and CONNECTED). The telephone system must also support dial by URL, i.e. it must be possible to call up a URL and thereby inform the telephone system that a certain destination number has been called by the phone at a workstation.

14.2.1.2 Incoming calls

The telephone system informs KIX of an action by calling up a URL (e.g. INCOMING):

<FQDN of KIX Server>/cti.pl?Action=<Action>&Src=<ID>&Dest=<ID>

e.g.: http://.../cti.pl?Action=INCOMING&Src=03715347620&Dest=1234

This URL would be called up, for example, for an incoming call from the number 0371/5347-620 to the internal number (extension) 1234. The destination number can also contain the complete sequence of numbers (incl. country code, area code, and main number).

Fundamentally every type of ID (IP, MAC, SIP, phone number,...) is supported as long as this ID is saved on the applicable dataset (customerID/employee) in KIX. Different types of IDs can be used for each contact person and agent.

Support for the following actions/events (designation can vary depending on the telephone system) is the best:

INCOMING	A caller is calling.
MISSED	The phone call was missed.
CONNECTED	The call is connected by the telephone system to a workplace telephone, i.e. the employee has picked up.
HANGUP	The phone call has ended.

14.2.1.3 Outgoing calls

So that KIX can automatically archive the outgoing phone calls of employees, and assign them to the applicable processes, the telephone must support dial by URL. This means that when dialing a certain phone number, calling up a URL is sufficient. This URL contains the phone number/ID of the employee's phone and the desired destination phone number/ID.
The CTI module also supports a direct dial by URL via the employee's telephone. Here the IP of the applicable telephone must be available in the dataset of the employee.

After KIX has called up the URL, the telephone system dials the destination phone number and connects the call to the phone of the applicable employee.

Example (server dial):	<FQDN of telephone system server>/dialOut?Src=1234&Dest=03715347620
Example (telephone dial):	<FQDN of telephone>/dialOut?Number=03715347620

The structure of the URL is specified by the telephone system and can be configured in KIX. Here, KIX can also use a connection via a proxy server (incl. authentication) for the server dial out.

14.2.2 Package Installation

Install the package "kixpro-cti" using "Paketverwaltung (Package Administration)" (Menu "Admin" area "System Administration") or using the command line.

14.2.3 Integration in Your Telephone System

CTI4KIX already has a CTI backend. This supports communication between KIX Professional and your telephone system by using HTTP requests.

14.2.4 HTTP Backend – Configuration of ActionURLs

To be able to use the HTTP backend and all CTI functions, your telephone system must support so-called "ActionURLs". The required ActionURLs must be configured to your phone so that communication with the CTI4KIX add-on is possible.

For each required ActionURL, this URL must be configured:

<FQDN of KIX Server>/cti.pl?Action=<Action>&Src=<ID>&Dest=<ID>

The parameter "ID" can be any identifier (MAC, SIP, telephone number) as long as the identifier is identical to that in the contact entry of an agent or a contact person. The parameter "Action" can be one of the following:

INCOMING	An incoming telephone call has been registered in your telephone system.
MISSED	An incoming telephone call was not connected to a telephone, and the caller has hung up.
CONNECTED	An incoming telephone call has been connected to a telephone.
HANGUP	The telephone call has been ended.

To identify the correct customer contact and the destination of an incoming call, the action CONNECTED must at least be configured. If this communication is correctly configured and works, an agent can open the input screen for a new ticket once they have picked up the phone. The information about the caller and all corresponding information (for example, assigned ConfigItems) is shown at this stage if it is possible to find this in the customer database.

Please note

If you use another CTI backend, please read the documentation for this backend for further information.

15 Time Accounting

15.1 Installation of Time Accounting Add-On Module

15.1.1 Prerequisites

- KIX Professional 17 – current version

15.1.2 Package Installation

Install the package "KIXTicketAccounting" using "Paketverwaltung (Package Administration)" (Menu "Admin" area "System Administration") or using the command line.

15.2 Configuration of Time Accounting Add-On Module

<table>
<tr><th>Begriff</th><th>Erklärung</th></tr>
<tr><td>PreferencesGroups###PreferencesTimeAccounting</td><td>Registration of a new personal setting.</td></tr>
<tr><td>Frontend::Module###AgentTicketTimeAccountingEdit</td><td>Frontend module registration.</td></tr>
<tr><td>Frontend::Module###AgentTicketZoomTabTimeAccounting</td><td>Frontend module registration</td></tr>
<tr><td>Frontend::Output::FilterElementPre###TimeAccountingInputStyleHHMM</td><td>De-/activates output filter for the layout change for entering time in format hh:mm<table><tr><td>"TemplateFileRegexp"</td><td>Permits restriction to certain tt files</td></tr><tr><td>"ReferToEditTimeTabRegexp"</td><td>Regular expression which includes all ticket templates which permit the editing of time accounting.
Posted times must always be edited in the time accounting tab.</td></tr></table></td></tr>
<tr><td>Frontend::Output::FilterElementPre###TimeAccountingTimeUnitsCheck</td><td>De-/activates output filter for determining the minimum time to be posted.<table><tr><td>"MinimumTimeUnits"</td><td>Specifies the minimum time to be entered in minutes.</td></tr></table></td></tr>
<tr><td>Loader::Agent::CommonJS###900-KIXTicketTimeAccounting</td><td>Registration of JS-Dateien.</td></tr>
</table>

Begriff	Erklärung
Loader::Agent::CommonCSS###900-KIXTicketTimeAccounting	Registration of CSS-Dateien.
Ticket::Acl::Module###500-AccountetTimeRestrictTicketStates	Ticket ACL to restrict ticket states and ticket actions for tickets without time entry.
Ticket::CustomModule###900-KIXTicketTimeAccounting	Expansion of thetTicket function.
Ticket::Frontend::AgentTicketTimeAccountingEdit###Permission	Required permision.
Ticket::Frontend::AgentTicketZoomTabAccountedTime###AccountedTimeArticle	When the working time is recorded without article, should these time units then be attached to an article?
Ticket::Frontend::AgentTicketZoomTabTimeAccounting###RedirectURL	URL for redirecting when popup is closed.
TicketTimeAccounting::ReferenceFields###InitialReference	Specification of the pre-selected posting reference *"Ticket Owner Attribute"* – Personal preference of ticket owner. *"Ticket Responsible Attribute"* – Personal preference of person responsible for the ticket *"Current User Attribute"* – Personal preference of agent. *"Ticket Attribute"* – Preference saved on the ticket

<table>
<tr><th>Begriff</th><th>Erklärung</th></tr>
<tr><td>TicketTimeAccounting::ReferenceFields::1</td><td>Configuration for the first reference field:<table>
<tr><td>"DynamicField"</td><td>Dynamic field for reference (article based)</td></tr>
<tr><td>"PreferenceDynamicField"</td><td>Dynamic field for preference (ticket based)</td></tr>
<tr><td>"Mandatory"</td><td>Mandatory field setting for "AgentTicketTimeAccountingEdit" and "AgentTicketZoomTabTimeAccounting"</td></tr>
<tr><td>"ShowInStats"</td><td>Use reference in statistics</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Begriff</th><th>Erklärung</th></tr>
<tr><td>TicketTimeAccounting::ReferenceFields::2</td><td>Configuration for the second reference field:<table>
<tr><td>"DynamicField"</td><td>Dynamic field for reference (article based)</td></tr>
<tr><td>"PreferenceDynamicField"</td><td>Dynamic field for preference (ticket based)</td></tr>
<tr><td>"Mandatory"</td><td>Mandatory field setting for "AgentTicketTimeAccountingEdit" and "AgentTicketZoomTabTimeAccounting"</td></tr>
<tr><td>"ShowInStats"</td><td>Use reference in statistics</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Begriff</th><th>Erklärung</th></tr>
<tr><td>TicketTimeAccounting::ReferenceFields::3</td><td>Configuration for the third reference field:<table>
<tr><td>"DynamicField"</td><td>Dynamic field for reference (article based)</td></tr>
<tr><td>"PreferenceDynamicField"</td><td>Dynamic field for preference (ticket based)</td></tr>
<tr><td>"Mandatory"</td><td>Mandatory field setting for "AgentTicketTimeAccountingEdit" and "AgentTicketZoomTabTimeAccounting"</td></tr>
<tr><td>"ShowInStats"</td><td>Use reference in statistics</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Begriff</th><th>Erklärung</th></tr>
<tr><td>TicketTimeAccounting::ReferenceFields::4</td><td>Configuration for the fourth reference field:<table>
<tr><td>"DynamicField"</td><td>Dynamic field for reference (article based)</td></tr>
<tr><td>"PreferenceDynamicField"</td><td>Dynamic field for preference (ticket based)</td></tr>
<tr><td>"Mandatory"</td><td>Mandatory field setting for "AgentTicketTimeAccountingEdit" and "AgentTicketZoomTabTimeAccounting"</td></tr>
<tr><td>"ShowInStats"</td><td>Use reference in statistics</td></tr>
</table></td></tr>
<tr><td>TicketTimeAccounting::Flag</td><td>Configuration for the flag field:<table>
<tr><td>"DynamicField"</td><td>Dynamic field for reference (article based)</td></tr>
<tr><td>"Mandatory"</td><td>Mandatory field setting for "AgentTicketTimeAccountingEdit" and "AgentTicketZoomTabTimeAccounting"</td></tr>
<tr><td>"ShowInStats"</td><td>Use reference in statistics</td></tr>
</table></td></tr>
</table>

Begriff	Erklärung
Ticket::Frontend::AgentTicketZoomTabTimeAccounting###TableFields	Configuration of the table columns to be displayed. Default configuration includes all possible entries: 100::Date → Date 110::Time Units → TimeUnitsDHMS 120::Editor → CreateBy 200::Project → ReferenceField 1 210::Subproject → ReferenceField2 220::Workpackage → ReferenceField3 230::Subworkpackage → ReferenceField4 300::Description → Description Key consists of numerical prefix + "::" + column header that defines the column sequence. The value identifies the applicable attribute to be shown.
Ticket::EventModulePost###CreateKIXTTAEntry	Registration of ticket event.
DashboardBackend###0300-TimeAccounting	Frontend module registration.
Frontend::Module###AgentITSMConfigItemAccountedTickets	Registration of frontend modules for display and CSV download of the accounted tickets.
ITSMConfigItem::Frontend::MenuModule###900-AccountedTickets	Activation of CI action "Accounted Tickets" when viewing an ITSM Config Item.
ITSMConfigItem::Frontend::AgentITSMConfigItemAccountedTickets###Permission	Required permission to call up function. Default: "RO".
ITSMConfigItem::Frontend::AgentITSMConfigItemAccountedTickets###Selection	Registration of the sections contained in action "AccountedTickets" • AccountedTickets::Section001 • AccountedTickets::Section002

Begriff	Erklärung
AccountedTickets::Section001###-Config	Configuration for section • Title = Accounted Tickets • Description = Shows ticket which account time to current object. • KIXTTAReference = 1 • LinkReference = 0 • LinkType = ReworkFor
AccountedTickets::Section001###-TicketList	Defines which ticket attributes are shown in the ticket list of the area. The column sequence is defined by alphanumeric sorting of the keys. For this, a prefix separated using a double colon is used. The prefix is followed by the tech. designation of the ticket attribute to be shown. The value contains the label of the column. • 001::TicketNumber = Ticketnumber • 010::Title = Title • 020::State = State • 030::Owner = Owner • 040::ObjectAccountedTime = Accounted Time (on this reference)
AccountedTickets::Section001###-DefaultSelection	Defines whether section in the area "Download CSV" is already marked or whether it has to be manually selected. • Standard: Yes
AccountedTickets::Section002###-Config	Siehe AccountedTickets::Section001###-TicketList • Title = Accounted Tickets • Description = Shows ticket which are not accounted but linked to current object • KIXTTAReference = 1 • LinkReference = 01 • LinkType = ReworkFor

Begriff	Erklärung
AccountedTickets::Section002###-TicketList	See AccountedTickets::Section001###-TicketList • Title = Accounted Tickets • Description = Shows ticket which are not accounted but linked to current object • KIXTTAReference = 1 • LinkReference = 01 • LinkType = ReworkFor
AccountedTickets::Section002###-DefaultSelection	See AccountedTickets::Section002###-DefaultSelection • Standard: Yes
LinkObject::Type###ReworkFor	LinkType registration • SourceName → Rework For • TargetName → Rework
LinkObject::PossibleLink###3605	• Activation of link Ticket ↔ ITSMConfigItem • Object1 → Ticket • Object2 → ITSMConfigItem • Type → ReworkFor
Frontend::KIXSidebarBackend###-BudgetOverview	Parameter for KIXSidebar backend BudgetOverview • Only reference level 1 (Project)
Frontend::KIXSidebarBackend###-BudgetDetailview	Parameter for KIXSidebar backend BudgetDetailview • Up to reference level 4 (Subworkpackage)
AgentCustomerInformationCenter::Backend###0800-CIC-BudgetOverview	Parameter for CIC dashboard BudgetOverview • Only reference level 1 (Project)
AgentCustomerInformationCenter::Backend###0805-CIC-BudgetOverview	Parameter for CIC dashboard BudgetOverview • Up to reference level 4 (Subworkpackage)

15.2.1 CIC Dashlet and Ticket Sidebar Free Budgets

Both features serve to display the already posted time allocation and time allocation still available as part of the project, sub-project, work package, or sub-work package referenced in each case. The functions are activated during package installation but are only available when the following conditions are met:

- the first posting reference is activated for time recording, making use of dynamic field type "{} ITSMConfigItemReference"

AND

- the definitions for CI classes "Project" and "Workpackage" are adapted as applicable MANUALLY. For the definitions of "Project" and "Workpackage", the attribute "CustomerCompany" (CustomerID) still has to be entered manually. "CustomerCompany CIACCustomerCompany" should be used as the attribute type.

```
{
Key => 'CustomerCompany',
Name => 'CustomerCompany',
Searchable => 1,
Input =>
    {
    Type => 'CIACCustomerCompany',
    },
}
```

The definition of "Project" is to be supplemented by one attribute "PlannedEffort" (required - Array[0..1]).

```
{
Key => 'PlannedEffort',
Name => 'Planned effort (hours),
Searchable => 0,
Input =>
    {
    Type => 'Text',
    Required => 1,
    Size => 50,
    MaxLength => 100,
    },
}
```

The times for work packages are listed with the default configuration (1 line per WP).

- Ticketsidebar: All (and only) work packages are listed that were posted in the ticket.
- CIC Dashlet: All (and only those) work packages are listed that are allocated to the Customer ID AND were posted in at least one ticket.

It is possible to configure whether the times for projects, sub-projects, work packages or sub-work packages should be listed (possibly for CIs of other classes depending on the configuration of posting references).

Available default columns are:

- Project
- Sub-project
- Work package
- posted
- available

Here the number "posted" is determined from ALL postings to this work package (the sum of postings to the WP from ALL tickets).

The number "available" is the difference between "posted" and the value in (configurable) CI class attribute _"PlannedEffort".

If the value in "available" is negative, the line is marked red. The display of columns project/sub-project/work package/sub-work package can be configured.

16 Liability Disclaimer for Add-On Modules

16.1 Liability for Contents

The contents of our pages and documents have been prepared with the utmost care. Nevertheless, no liability can be accepted for any technical or editorial errors or omissions in this document. This also applies to any incidental or consequential damages that may arise from the provision, function or use of this material.

Please feel free to send any comments regarding design, additions or possible errors to our support team (https://forum.kixdesk.com) at any time. We will gladly take up and implement sensible suggestions and improvements.

All data, features and descriptions given in this work are subject to change at any time and without notice. Personal names and company names are fictitious. Any coincidences with real persons and companies are purely coincidental.

As a service provider, we are responsible for our own content on these pages in accordance with the general laws pursuant to § 7 para. 1 German Teleservices Act. However, according to §§ 8 to 10 German Teleservices Act, we are not obliged as a service provider to monitor transmitted or stored third-party information or to investigate circumstances that indicate illegal activity.

Obligations to remove or block the use of information in accordance with general laws remain unaffected by this. However, liability in this respect is only possible from the point in time at which a concrete infringement of the law becomes known. If we become aware of any such infringements, we will remove the relevant content immediately.

16.2 Liability for Links

Our website contains links to external websites of third parties over whose content we have no influence. Therefore, we do not assume any liability for these external contents. The respective provider or operator of the pages is always responsible for the content of the linked pages. The linked pages were checked for possible legal violations at the time of linking. Illegal contents were not recognisable at the time of linking.

A permanent control of the contents of the linked pages is not reasonable without concrete evidence of a violation of the law. If we become aware of any infringements of the law, we will remove such links immediately.

16.3 Copyright

The contents created by the site operators are subject to German copyright law. Duplication, processing, distribution, or any form of commercialization of such material beyond the scope of the copyright law shall require the prior written consent of its respective author or creator.

Insofar as the content on this site was not created by the operator, the copyrights of third parties are respected. In particular, third-party content is identified as such. Should you nevertheless become aware of a copyright infringement, please inform us accordingly. If we become aware of any infringements, we will remove such content immediately.

17 Purpose for which the use of KIX add-on modules is intended within a medical context

KIX add-on modules is not suitable, intended or approved for the identification, prevention, monitoring, treatment, relief or compensation of illnesses, injuries and disabilities. KIX add-on modules is also not intended for the examination, replacement or modification of the anatomical structure or of a physiological process. KIX add-on modules must also not be used to directly control a diagnostic or therapeutic product. Furthermore, KIX add-on modules is not designed to be used in conjunction with a medical device with the intention of enabling the medical device to fulfill its intended purpose.

In a medical context, KIX add-on modules is exclusively intended for carrying out administration and database-related tasks. If the above restrictions are complied with, KIX add-on modules is suitable for use in a medical environment

- purely for documentation purposes, such as:
  - the general management of equipment in the form of managing and cataloging device data (device meta data) such as names, IP addresses, series numbers, persons responsible, guarantee periods, service providers, operating documents, license information, cost centers, as well as the management/organisation of users, device instructions;
  - the central documentation of all activities and changes in the IT such as due to executed maintenance activities or other service activities (e.g. medical device log book);
  - for compiling a knowledge database.
- for automating and simplifying general management processes, such as:
  - in service and technical customer service, for example in IT service (errors, changes, maintenance);
  - in building services (errors, changes, cleaning) or medical device technology.
- for monitoring purposes and calendar functions, such as:
  - for central IT services (network, email, data servers, SAP,...);
  - and for error and requirement notifications for the IT team, building services, medical device technology;
  - for the planning of regular maintenance works and reminders for replacing wear parts;
  - for the organisation of regular orders and planning the deployment of service technicians.

KIX add-on modules is not designed for enabling or guaranteeing the functioning of medical devices and must therefore not be used for these purposes. If in the context of the aforementioned functions KIX add-on modules also allows data exchange via an interface, please note that KIX add-on modules must not be used for data modification or for any type of data control for medical or therapeutic purposes.

KIX add-on modules may only be used in a medical context within the approved parameters mentioned above. KIX Service Software GmbH assumes no liability for any use that goes beyond or deviates from the approved parameters.