# OTOBO Installation Guide

## Release rel-11.0

Rother OSS GmbH

Nov 14, 2024

# Table of Contents

Overview

## 1.1 Summary

With this documentation you should gain a good overview about the OTOBO package "ITSMConfiguration Management". We provide a basic configuration based on a selection of must usual CIs to get you up and running quickly. If you are interested in getting your own CIs configured you should be able to do this too with the adavanced configuraion.

## 1.2 Introduction

With this package you can extend the functionality in OTOBO to manage assets and CIs and link them to tickets, customer users, etc. to keep track of thoose and have an combined solution for your tickets and assets.

## 1.3 Common Usecases

- When a support agent receives an incident ticket, the CMDB allows them to view relevant CIs directly associated with the affected systems or users. They can quickly check for recent changes, dependencies, and known issues.

- When planning a system change, the CMDB integration can help teams analyze potential impacts by displaying dependencies between systems, applications, and infrastructure components.

- The lifecycle of assets and CIs, such as computers, software, or network devices can be tracked within OTOBO's CMDB.

- For recurring or complex issues, support teams can use the CMDB to view the historical data and relationships between affected CIs, assisting in identifying root causes.

- For organizations with regulatory or compliance requirements, a CMDB provides a single source of truth for asset configurations and changes.

## 1.4 Further Reading

If you are unfamiliar with the concept of a CMDB you might want to check out the article on wikipedia <https://en.wikipedia.org/wiki/Configuration_management_database>.

Configuration

## 2.1 System requirements

### 2.1.1 Framework

OTOBO 11.0.x

### 2.1.2 Packages

ITSMCore 11.0.1

### 2.1.3 Third-party software

-

## 2.2 Basic Configuration

Following the basic configuration you will learn to assign permissions to agents and customer users and how to get started with the most usual CIs.

### 2.2.1 Assign Permissions

Agents/customer users who need access to CIs of the CMDB have to be assigned to the group "itsm-configitem".

### 2.2.2 Starting with the most usual CIs

OTOBO provides the option to import class bundles showcasing the most usual CIs. To start with them please perform the following actions:

1. Open the admin-view of your OTOBO Web-UI.

2. Open "Config Items" in the section "CMDB settings".

3. Locate the "Ready2Import Class Bundles" panel on the left side and select the class bundle you want to import.
   .. figure:: images/ready2import_class_bundles.png
         :alt: Ready2Import Class Bundles panel

4. Click the button to import the Ready2Adopt class bundles.

## 2.3 Custom Configuration and Advanced Features

### 2.3.1 Classes and Roles in the General Catalog

### 2.3.2 Config Item Definitions

**General Structure**

Example class "Domain":

```
Pages:
# The individual tabs in AgentITSMConfigItemZoom are called Pages.
# The fields to be displayed there are defined in the roles
# listed on the left under "Roles".
  - Name: Summary
    Layout:
       Columns: 2
       ColumnWidth: 1fr 1fr
    Interfaces:
       - Agent
    Content:
       - Section: General Fields::Domain Information
         RowStart: 1
         ColumnStart: 1
       - Section: CMDB Backlinks::Backlinks
         RowStart: 2
         ColumnStart: 1
       - Section: CMDB Description::Description
         RowStart: 1
         RowSpan: 2
         ColumnStart: 2
  - Name: Accounting
    Layout:
       Columns: 2
       ColumnWidth: 1fr 1fr
    Interfaces:
       - Agent
    Content:
       - Section: CMDB Accounting::Accounting
         RowStart: 1
```

```
            ColumnStart: 1
Sections:
   Accounting:
      Content:
         -  Header: Accounting
         -  DF: ITSMAccounting-InventoryNumber
         -  Grid:
               Columns: 3
               ColumnWidth: 1fr 1fr 1fr
               Rows:
                  -
                     - DF: ITSMAccounting-DateOfInvoice
                     - DF: ITSMAccounting-OrderDate
                     - DF: ITSMAccounting-WarrantyDate
                  -
                     - DF: ITSMAccounting-CostUnit
                     - DF: ITSMAccounting-InvestmentCosts
                     - DF: ITSMAccounting-OperatingCosts
                  -
                     - DF: ITSMAccounting-OrderNumber
                     - DF: ITSMAccounting-InvoiceNumber
                     - DF: ITSMAccounting-ReferenceToAccount
Roles:
   # Roles are used to define the Sections and their individual fields.
   # Of course, sections can also be defined directly here in the class definitions.
   # We have simply moved them to "Roles" for the sake of clarity.

   # For each class with class-specific fields, there is a "General Fields"
   # definition referring to a role with the name "ITSM GF class name".
   # This is where the fields of the respective class can be customised.
   General Fields:
      Name: ITSM Domain Information
```

## Setting Reference

### Pages

- Name (required) - The name of the page

- Layout (required) - The layout of the grid on this page

  - **Columns - number of columns**  e.g. Columns: 3 will display 3 columns

  - ColumnWidth - basic css defining the column widths

- Interfaces (optional) - An array containing the interfaces on which this page is available (default: [Agent])

- Groups (optional) - If defined, an array containing groups which a user has to be member of, to be able to view this page

- Content (required) - The content, i.e. an array of sections and their position on this page

  - Section (required) - The section name

  - RowStart (optional) - The starting row in the page grid of this section

  - RowSpan (optional) - The number of rows this section should span

  - ColumnStart (optional) - The starting column in the page grid of this section

– ColumnSpan (optional) - The number of columns this section should span

**Sections** The Sections section of the yaml contains a hash of the Sections which can be referenced in the Pages. Under each of those sections the following keys are valid: * Type (optional) - Defines the type of the section. Depending on the type other attributes might be available or not. Currently available are:

- DynamicFields (default) - A standard section containing dynamic fields
- Description - A Richtext description possibly containing images etc., which can be defined on the CI edit masks
- ConfigItemLinks - Show ConfigItems linked via dynamic fields (not used for edit masks)
- ReferencedSection - Show a section of a CI referenced in a reference dynamic field (not used for edit masks)

**Type: DynamicFields** An additional key Content is mandatory here. The Content is essentially the same as in ticket masks. Additionally a header for the section can be provided. * Header (optional) - A header for this section * DF - A dynamic field (the name)

- Mandatory (optional) - 1 if in edit masks the field should be mandatory
- Readonly (optional) - 1 if in edit masks the field should be readonly (only for basic field types)
- Label (optional) - Override the label of the field on edit masks
- Grid - A multi column section of dynamic fields
    - Columns - Number of columns
    - ColumnWidth (optional) - Widths of the colums (e.g. "1fr 40px 2fr"; "%" cannot be used)
    - Rows - The dynamic field matrix (i.e. an array of arrays [[DF: DF11,DF: DF12, DF: DF13], [DF: DF21, DF: DF22, DF: DF23], ...])

Each row mus look as follows:

**- (The indicator for a each line in the grid as array)**

- DF - A dynamic field (the name - see above)
    - ColumnStart
    - ColumnSpan

Example for a dynamic field section (the Type is implicit, if none is given):

```
Sections:
   Info:
      Content:
         - DF: Computer-OS
         - DF: Owner
         Mandatory: 1
         Label: In front of the monitor
         - Grid:
            Columns: 2
            ColumnWidth: 1fr 1fr
            Rows:
               # The first row of the grid
               -
                  - DF: DateBought
                    Readonly: 1
                  - DF: DateWarranty
               # The second row of the grid
```

```
          -
              - DF: Computer-Application
              - DF: Computer-LicenseKey
```

**Type: Description** No further settings are available.

**Type: ConfigItemLinks** Lists linked config items. * Header (optional) - A header for this section * LinkedAs (optional) - Source (default), Target, Both * LinkTypes (optional) - An array containing link types

**Type: ReferencedSection** * ReferenceField (required) - The reference field of this config item, containing the referenced config item * SectionName (required) - The name of the section of the referenced config item which should be shown * FieldListPre (optional) - only valid if the referenced section is of type DynamicField (or ReferencedSection, referencing a DynamicField section): Dynamic fields of this Config Item to be rendered in front of the dynamic fields of the referenced section * FieldListPost (optional) - see above, but after the referenced dynamic field section

**Type: Module** This is not yet implemented * Module (required) - A custom module returning custom HTML to be rendered in this section

## Config Item Versioning and History

## Dynamic Fields specifics

Fields for CI classes are defined as Dynamic Fields with a new Object Type ITSM ConfigItem. To use fields, you have to define them as a Dynamic Fields for the CI Object Type and then name them in your CI Class Definition.

The following Dynamic Fields might be specifically interesting for your CMDB. * Reference

- Agent

  - With this field you can refer to an Agent in OTOBO.

  - This might be valuable to define the responsible of a ConfigItem, e.g. the IT Administrator who is also an Agent.

- ConfigItem

  - With this field you can refer to another ConfigItem in OTOBO.

  - This might be valuable to link multiple ConfigItems together.

  - This might be valuable to define a hirarchy structure, e.g. Country <-> Subsidiary <-> Building <-> Room>

- ConfigItemVersion

  - With this field you can refer to a specific Version of another ConfigItem in OTOBO.

- Customer

  - With this field you can refer to another Customer in OTOBO.

  - This might be valuable to define the Company or Department as connected to the ConfigItem.

  - With this, you are able to see the ConfigItem in the Widget of the Customer Information Center.

- Customer User

  - With this field you can refer to another Customer User in OTOBO.

- – This might be valuable to define the owner of a ConfigItem, e.g. if it is an internal Customer User (employee).
  - – With this, you are able to see the ConfigItem in the Widget of the Customer User Information Center.
- Dynamic Field Lens
  - – The idea of this field is to have a referenced object from which you want to see (like through the lens of a periscope?) specific values.
  - – In case you have a reference to a Configuration Item (via a DF Reference ConfigItem), you can look at or change values from that referenced Config Item via the Lens field. It might make sense to use the Readonly functionality to not be able to change those values.
  - – It is necessary to have referenced object.
  - – The referenced DF: This setting is to define, in which Dynamic Field the referred object is mentioned. This is the Config Item, of which I want to display the value.
  - – attribute DF of the referenced object: This is the (Dynamic) Field in this refered object, from which I want to see the value.
  - – I can make this field read only by using the "Readonly: 1"
  - – For example, you could refer to a Server ConfigItem and have a look at the IP Address and the Server Owner via the Lens fields.
- General Catalog
  - – The idea is to have a Dropdown field.
  - – You define the values for the General Catalog field in Admin > General Catalog.
  - – As a differentiation to a Dynamic Field of type "Dropdown" is, that you don't define the values per Dynamic Field, but per General Catalog.
  - – So you can use multiple General Catalog Dynamic Fields which all take the same Catalog as a value base.
  - – For example, you could have a two fields with the same Dropdown values, but with a different Name/Label of the Dynamic Field. E.g. a field named "INSERT EXAMPLE HERE" and another field named "INSERT EXAMPLE HERE" in different CI Classes, both with the same values.

## Config Item Links

Backlinks Tree View

## Import/Export

- CSV
- Automization possible via CronJob & bin/otobo.Console.pl Admin::ITSM::ImportExport::Import
- Alternative: Web Service?

## Console Commands

bin/otobo.Console.pl * Admin::ITSM::Configitem::ClassExport - Export dynamic field classes and their respective dynamic fields. * Admin::ITSM::Configitem::ClassImport - Import dynamic field classes and their

respective dynamic fields. * Admin::ITSM::Configitem::Delete - Delete config items (all, by class (and deployment state) or by number). * Admin::ITSM::Configitem::DumpGraph - Dump the graph of config items as SVG to the file OTOBO_ITSM_Config_Items.svg. * Admin::ITSM::Configitem::ListDuplicates - List ConfigItems which have a non-unique name. * Admin::ITSM::Configitem::UpgradeTo11 - Upgrade the complete CMDB from OTOBO 10 to OTOBO 11. All config item definitions will be changed, for each config item attribute a dynamic field will be prepared and the data will be migrated. * Maint::ITSM::Configitem::RebuildLinkTable - Rebuild the database table configitem_link from the dynamic fields of the type Reference. Only the fields linking config items are

### Permissions and Access Restrictions

- ACLs
    - New Object Type "ITSM ConfigItem"
    - Match settings
        * ConfigItem
        * Frontend
        * User
    - Change settings
        * ConfigItem
        * Form

### Migration from OTOBO 10 / OTRS

bin/otobo.Console.pl Admin::ITSM::Configitem::UpgradeTo11 - Upgrade the complete CMDB from OTOBO 10 to OTOBO 11. All config item definitions will be changed, for each config item attribute a dynamic field will be prepared and the data will be migrated.

Ready2Import Class Bundles - Wo? * IT-Servicemanagement

### Relations

When to use which? Which impact has which relation?

## 2.4 Configuration Reference

### 2.4.1 Core::BulkAction

#### ITSMConfigItem::Frontend::BulkFeature

Enables configuration item bulk action feature for the agent frontend to work on more than one configuration item at a time.

#### ITSMConfigItem::Frontend::BulkFeatureGroup

Enables configuration item bulk action feature only for the listed groups.

## 2.4.2 Core::DynamicFields::DriverRegistration

### DynamicFields::Driver###ConfigItem

DynamicField backend registration.

### DynamicFields::Driver###ConfigItemVersion

DynamicField backend registration.

## 2.4.3 Core::DynamicFields::ObjectTypeRegistration

### DynamicFields::ObjectType###ITSMConfigItem

DynamicField object registration.

## 2.4.4 Core::Elasticsearch::Settings

### Elasticsearch::ExcludedCIClasses

ConfigItems of the following classes will not be stored on the Elasticsearch server. To apply this to existing CIs, the CI migration has to be run via console, after changing this option.

### Elasticsearch::ExcludedCIDeploymentStates

ConfigItems with the following deployment states will not be stored on the Elasticsearch server. To apply this to existing CIs, the CI migration has to be run via console, after changing this option.

### Elasticsearch::ConfigItemStoreFields

Fields stored in the configuration item index which are used for other things besides fulltext searches. For the complete functionality all fields are mandatory.

### Elasticsearch::ConfigItemSearchFields

Fields of the configuration item index, used for the fulltext search. Fields are also stored, but are not mandatory for the overall functionality. Inclusion of attachments can be disabled by setting the entry to 0 or deleting it.

### Elasticsearch::QuickSearchShow###ConfigItem

Objects to search for, how many entries and which attributs to show. ConfigItem attributes have to explicitly be stored via Elasticsearch.

## 2.4.5  Core::Event::ITSMConfigItem

**ITSMConfigItem::EventModulePost###100-History**

Config item event module that enables logging to history in the agent interface.

**ITSMConfigItem::EventModulePost###300-DefinitionConfigItemUpdate**

Config item event module that updates config items to their current definition.

**ITSMConfigItem::EventModulePost###400-LinkSync**

Config item event module that updates the table configitem_Ĺink.

**ITSMConfigItem::EventModulePost###500-RecalcCurInciState**

Config item event module updates the current incident state.

**ITSMConfigItem::EventModulePost###4000-ScriptDynamicFields**

Evaluate all script fields.

**DynamicField::EventModulePost###200-DynamicFieldSync**

Dynamic field event module that marks config item definitions as out of sync, if containing dynamic fields change.

**ITSMConfigItem::EventModulePost###1000-GenericInterface**

Performs the configured action for each event (as an Invoker) for each configured Webservice.

## 2.4.6  Core::Event::Ticket

**ITSMConfigItem::EventModulePost###042-ITSMConfigItemTicketStatusLink**

Event module to set configitem-status on ticket-configitem-link.

**Ticket::EventModulePost###042-ITSMConfigItemTicketStatusLink**

Event module to set configitem-status on ticket-configitem-link.

## 2.4.7  Core::GeneralCatalog

**GeneralCatalogPreferences###DeploymentStates**

Parameters for the deployment states in the preferences view of the agent interface.

**GeneralCatalogPreferences###DeploymentStatesColors**

Parameters for the deployment states color in the preferences view of the agent interface.

**GeneralCatalogPreferences###NameModule**

Parameters for the name module for config item classes in the preferences view of the agent interface.

**GeneralCatalogPreferences###NumberModule**

Parameters for the number module for config item classes in the preferences view of the agent interface.

**GeneralCatalogPreferences###VersionStringModule**

Parameters for the version string module for config item classes in the preferences view of the agent interface.

**GeneralCatalogPreferences###VersionStringExpression**

Parameters for the version string template toolkit module for config item classes in the preferences view of the agent interface.

**GeneralCatalogPreferences###VersionTrigger**

Parameters for the version trigger for config item classes in the preferences view of the agent interface.

**GeneralCatalogPreferences###Categories**

Parameters for the categories for config item classes in the preferences view of the agent interface.

**GeneralCatalogPreferences###Permissions**

Parameters for the example permission groups of the general catalog attributes.

## 2.4.8 Core::ITSMConfigItem

**ITSMConfigItem::CINameRegex**

Defines regular expressions individually for each ConfigItem class to check the ConfigItem name and to show corresponding error messages.

**ITSMConfigItem::Permission::Class###010-ClassGroupCheck**

Module to check the group responsible for a class.

### ITSMConfigItem::Permission::Item###010-ItemClassGroupCheck

Module to check the group responsible for a configuration item.

### ITSMConfigItem::Hook

The identifier for a configuration item, e.g. ConfigItem#, MyConfigItem#. The default is ConfigItem#.

### UniqueCIName::EnableUniquenessCheck

Enables/disables the functionality to check ITSM onfiguration items for unique names. Before enabling this option you should check your system for already existing config items with duplicate names. You can do this with the console command Admin::ITSM::Configitem::ListDuplicates.

### UniqueCIName::UniquenessCheckScope

Check for a unique name only within the same ConfigItem class ('class') or globally ('global'), which means every existing ConfigItem is taken into account when looking for duplicates.

### Customer::ConfigItem::PermissionConditions###01

Define a set of conditions under which a customer is allowed to see a config item. Conditions can optionally be restricted to certain customer groups. Name is the only mandatory attribute. If no other options are given, all config items will be visible under that category.

### Customer::ConfigItem::PermissionConditions###02

Define a set of conditions under which a customer is allowed to see a config item. Conditions can optionally be restricted to certain customer groups. Name is the only mandatory attribute. If no other options are given, all config items will be visible under that category.

### Customer::ConfigItem::PermissionConditions###03

Define a set of conditions under which a customer is allowed to see a config item. Conditions can optionally be restricted to certain customer groups. Name is the only mandatory attribute. If no other options are given, all config items will be visible under that category.

### Customer::ConfigItem::PermissionConditions###04

Define a set of conditions under which a customer is allowed to see a config item. Conditions can optionally be restricted to certain customer groups. Name is the only mandatory attribute. If no other options are given, all config items will be visible under that category.

### Customer::ConfigItem::PermissionConditions###05

Define a set of conditions under which a customer is allowed to see a config item. Conditions can optionally be restricted to certain customer groups. Name is the only mandatory attribute. If no other options are given, all config items will be visible under that category.

Customer::ConfigItem::PermissionConditionColumns###Default

Customer::ConfigItem::PermissionConditionColumns###01

Customer::ConfigItem::PermissionConditionColumns###02

Customer::ConfigItem::PermissionConditionColumns###03

Customer::ConfigItem::PermissionConditionColumns###04

Customer::ConfigItem::PermissionConditionColumns###05

CMDBTreeView::DefaultDepth

Defines the default relations depth to be shown.

CMDBTreeView::ShowLinkLabels

Defines if the link type labels must be shown in the node connections.

## 2.4.9 Core::ITSMConfigItem::ACL

ITSMConfigItemACLKeysLevel1Match

Defines which items are available in first level of the ITSM Config Item ACL structure.

ITSMConfigItemACLKeysLevel1Change

Defines which items are available in first level of the ITSM Config Item ACL structure.

ITSMConfigItemACLKeysLevel2::Possible

Defines which items are available in second level of the ITSM Config Item ACL structure.

ITSMConfigItemACLKeysLevel2::PossibleAdd

Defines which items are available in second level of the ITSM Config Item ACL structure.

ITSMConfigItemACLKeysLevel2::PossibleNot

Defines which items are available in second level of the ITSM Config Item ACL structure.

ITSMConfigItemACLKeysLevel2::Properties

Defines which items are available in second level of the ITSM Config Item ACL structure.

**ITSMConfigItemACLKeysLevel2::PropertiesDatabase**

Defines which items are available in second level of the ITSM Config Item ACL structure.

**ITSMConfigItemACLKeysLevel3::Actions###100-Default**

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

**ConfigItemACL::ACLPreselection**

Whether the execution of ConfigItemACL can be avoided by checking cached field dependencies. This can improve loading times of formulars, but has to be disabled, if ACLModules are to be used for ITSMConfigItem- and Form-ReturnTypes.

**ConfigItemACL::Autoselect**

Whether fields should be automatically filled (1), and in that case also be hidden from ticket formulars (2).

## 2.4.10  Core::ImportExport::ObjectBackend::ModuleRegistration

**ImportExport::ObjectBackendRegistration###ITSMConfigItem**

Object backend module registration for the import/export module.

## 2.4.11  Core::LinkObject

**LinkObject::DefaultSubObject###ITSMConfigItem**

Defines the default subobject of the class 'ITSMConfigItem'.

**LinkObject::ITSMConfigItem::ShowColumns**

Defines the shown columns of CIs in the link table complex view for all CI classes. If there is no entry, then the default columns are shown.

**LinkObject::ITSMConfigItem::ShowColumnsByClass**

Defines the shown columns of CIs in the link table complex view, depending on the CI class. Each entry must be prefixed with the class name and double colons (i.e. Computer::). There are a few CI-Attributes that common to all CIs (example for the class Computer: Computer::Name, Computer::CurDeplState, Computer::CreateTime). To show individual CI-Attributes as defined in the CI-Definition, the following scheme must be used (example for the class Computer): Computer::HardDisk::1, Computer::HardDisk::1::Capacity::1, Computer::HardDisk::2, Computer::HardDisk::2::Capacity::1. If there is no entry for a CI class, then the default columns are shown.

## 2.4.12 Core::LinkStatus

**ITSMConfigItem::SetIncidentStateOnLink**

Set the incident state of a CI automatically when a Ticket is Linked to a CI.

**ITSMConfigItem::LinkStatus::TicketTypes**

Defines which type of ticket can affect the status of a linked CI.

**ITSMConfigItem::LinkStatus::DeploymentStates**

Defines the relevant deployment states where linked tickets can affect the status of a CI.

**ITSMConfigItem::LinkStatus::IncidentStates**

Defines the order of incident states from high (e.g. cricital) to low (e.g. functional).

**ITSMConfigItem::LinkStatus::LinkTypes**

Defines which type of link (named from the ticket perspective) can affect the status of a linked CI.

## 2.4.13 Core::Stats

**Stats::DynamicObjectRegistration###ITSMConfigItem**

Module to generate ITSM config item statistics.

## 2.4.14 Daemon::SchedulerCronTaskManager::Task

**Daemon::SchedulerCronTaskManager::Task###TriggerConfigItemFetch-01**

Triggers ConfigItemFetch invoker automatically.

**Daemon::SchedulerCronTaskManager::Task###TriggerConfigItemFetch-02**

Triggers ConfigItemFetch invoker automatically.

**Daemon::SchedulerCronTaskManager::Task###TriggerConfigItemFetch-03**

Triggers ConfigItemFetch invoker automatically.

## 2.4.15 Frontend::Admin

**Events###ITSMConfigItem**

List of all Package events to be displayed in the GUI.

### 2.4.16 Frontend::Admin::ModuleRegistration

Frontend::Module###AdminITSMConfigItem

Frontend module registration for the agent interface.

Frontend::Module###AdminGenericInterfaceInvokerConfigItem

Frontend module registration for the agent interface.

### 2.4.17 Frontend::Admin::ModuleRegistration::AdminOverview

Frontend::NavigationModule###AdminITSMConfigItem

Admin area navigation for the agent interface.

### 2.4.18 Frontend::Admin::ModuleRegistration::Loader

Loader::Module::AdminITSMConfigItem###002-ITSMConfigItem

Loader module registration for the agent interface.

Loader::Module::AdminGenericInterfaceInvokerConfigItem###007-ITSMConfigurationManagement

Loader module registration for the agent interface.

### 2.4.19 Frontend::Admin::ModuleRegistration::MainMenu

Frontend::Navigation###AdminITSMConfigItem###003-ITSMConfigItem

Main menu item registration.

### 2.4.20 Frontend::Admin::View::ITSMConfigItemDefinition

ITSMConfigItem::Frontend::AdminITSMConfigItem###EditorRows

Defines the number of rows for the CI definition editor in the admin interface.

### 2.4.21 Frontend::Agent::ITSMConfigItem::MenuModule

ITSMConfigItem::Frontend::MenuModule###000-Back

Shows a link in the menu to go back in the configuration item zoom view of the agent interface.

### ITSMConfigItem::Frontend::MenuModule###200-History

Shows a link in the menu to access the history of a configuration item in the its zoom view of the agent interface.

### ITSMConfigItem::Frontend::MenuModule###300-Edit

Shows a link in the menu to edit a configuration item in the its zoom view of the agent interface.

### ITSMConfigItem::Frontend::MenuModule###400-Print

Shows a link in the menu to print a configuration item in the its zoom view of the agent interface.

### ITSMConfigItem::Frontend::MenuModule###500-Link

Shows a link in the menu that allows linking a configuration item with another object in the config item zoom view of the agent interface.

### ITSMConfigItem::Frontend::MenuModule###550-TreeView

Shows a link in the menu to display the configuration item links as a Tree View.

### ITSMConfigItem::Frontend::MenuModule###600-Duplicate

Shows a link in the menu to duplicate a configuration item in the its zoom view of the agent interface.

### ITSMConfigItem::Frontend::MenuModule###700-ConfigItemDelete

Shows a link in the menu to delete a configuration item in its zoom view of the agent interface.

## 2.4.22 Frontend::Agent::ITSMConfigItem::MenuModulePre

### ITSMConfigItem::Frontend::PreMenuModule###100-Zoom

Shows a link in the menu to zoom into a configuration item in the configuration item overview of the agent interface.

### ITSMConfigItem::Frontend::PreMenuModule###200-History

Shows a link in the menu to access the history of a configuration item in the configuration item overview of the agent interface.

### ITSMConfigItem::Frontend::PreMenuModule###300-Duplicate

Shows a link in the menu to duplicate a configuration item in the configuration item overview of the agent interface.

## 2.4.23 Frontend::Agent::ITSMConfigItem::Permission

**ITSMConfigItem::Frontend::AgentITSMConfigItem###Permission**

Required permissions to use the ITSM configuration item screen in the agent interface.

**ITSMConfigItem::Frontend::AgentITSMConfigItemEdit###Permission**

Required permissions to use the edit ITSM configuration item screen in the agent interface.

**ITSMConfigItem::Frontend::AgentITSMConfigItemAdd###Permission**

Required permissions to use the add ITSM configuration item screen in the agent interface.

**ITSMConfigItem::Frontend::AgentITSMConfigItemHistory###Permission**

Required permissions to use the history ITSM configuration item screen in the agent interface.

**ITSMConfigItem::Frontend::AgentITSMConfigItemPrint###Permission**

Required permissions to use the print ITSM configuration item screen in the agent interface.

**ITSMConfigItem::Frontend::AgentITSMConfigItemZoom###Permission**

Required permissions to use the ITSM configuration item zoom screen in the agent interface.

**ITSMConfigItem::Frontend::AgentITSMConfigItemSearch###Permission**

Required permissions to use the ITSM configuration item search screen in the agent interface.

**ITSMConfigItem::Frontend::AgentITSMConfigItemTreeView###Permission**

Required permissions to use the ITSM configuration item tree view screen in the agent interface.

**ITSMConfigItem::Frontend::AgentITSMConfigItemBulk###Permission**

Required permissions to use the bulk ITSM configuration item screen in the agent interface.

## 2.4.24 Frontend::Agent::ITSMConfigItemAttachment::Permission

**ITSMConfigItem::Frontend::AgentITSMConfigItemAttachment###Permission**

Required permissions to use the ITSM configuration item attachment action in the agent interface.

## 2.4.25 Frontend::Agent::ITSMConfigItemOverview

ITSMConfigItem::Frontend::Overview###Small

Defines an overview module to show the small view of a configuration item list.

## 2.4.26 Frontend::Agent::LinkObject

LinkObject::ComplexTable::SettingsVisibility###ITSMConfigItem

Define Actions where a settings button is available in the linked objects widget (LinkObject::ViewMode = "complex"). Please note that these Actions must have registered the following JS and CSS files: Core.AllocationList.css, Core.UI.AllocationList.js, Core.UI.Table.Sort.js, Core.Agent.TableFilters.js and Core.Agent.LinkObject.js.

## 2.4.27 Frontend::Agent::ModuleRegistration

Frontend::Module###AgentITSMConfigItem

Frontend module registration for the agent interface.

Frontend::Module###AgentITSMConfigItemZoom

Frontend module registration for the agent interface.

Frontend::Module###AgentITSMConfigItemEdit

Frontend module registration for the agent interface.

Frontend::Module###AgentITSMConfigItemPrint

Frontend module registration for the agent interface.

Frontend::Module###AgentITSMConfigItemDelete

Frontend module registration for the agent interface.

Frontend::Module###AgentITSMConfigItemTreeView

Frontend module registration for the agent interface.

Frontend::Module###AgentITSMConfigItemAdd

Frontend module registration for the agent interface.

**Frontend::Module###AgentITSMConfigItemSearch**

Frontend module registration for the agent interface.

**Frontend::Module###AgentITSMConfigItemHistory**

Frontend module registration for the agent interface.

**Frontend::Module###AgentITSMConfigItemBulk**

Frontend module registration for the agent interface.

**Frontend::Module###AgentITSMConfigItemAttachment**

Frontend module registration for the agent interface.

## 2.4.28 Frontend::Agent::ModuleRegistration::Loader

**Loader::Module::AgentITSMConfigItem###003-ITSMConfigItem**

Loader module registration for the agent interface.

**Loader::Module::AgentITSMConfigItemZoom###003-ITSMConfigItem**

Loader module registration for the agent interface.

**Loader::Module::AgentITSMConfigItemEdit###003-ITSMConfigItem**

Loader module registration for the agent interface.

**Loader::Module::AgentITSMConfigItemTreeView###003-ITSMConfigItem**

Loader module registration for the agent interface.

**Loader::Module::AgentITSMConfigItemAdd###003-ITSMConfigItem**

Loader module registration for the agent interface.

**Loader::Module::AgentITSMConfigItemSearch###003-ITSMConfigItem**

Loader module registration for the agent interface.

**Loader::Module::AgentITSMConfigItemHistory###003-ITSMConfigItem**

Loader module registration for the agent interface.

### 2.4.29 Frontend::Agent::ModuleRegistration::MainMenu

**Frontend::Navigation###AgentITSMConfigItem###003-ITSMConfigItem**

Main menu item registration.

**Frontend::Navigation###AgentITSMConfigItemAdd###003-ITSMConfigItem**

Main menu item registration.

**Frontend::Navigation###AgentITSMConfigItemSearch###003-ITSMConfigItem**

Main menu item registration.

**Frontend::Navigation###AgentITSMConfigItemBulk###003-ITSMConfigItem**

Main menu item registration.

### 2.4.30 Frontend::Agent::ModuleRegistration::MainMenu::Search

**Frontend::Search###ConfigItem**

Configuration item search backend router of the agent interface.

**Frontend::Search::JavaScript###ConfigItem**

JavaScript function for the search frontend.

### 2.4.31 Frontend::Agent::View::AgentITSMConfigItemBulk

**ITSMConfigItem::Frontend::AgentITSMConfigItemBulk###DynamicField**

Dynamic fields shown in the additional ITSM field screen of the agent interface.

### 2.4.32 Frontend::Agent::View::AgentITSMConfigItemEdit

**ITSMConfigItem::Frontend::AgentITSMConfigItemEdit###DynamicField**

Dynamic fields shown in the additional ITSM field screen of the agent interface.

### 2.4.33 Frontend::Agent::View::AgentITSMConfigItemHistory

**ITSMConfigItem::Frontend::AgentITSMConfigItemPrint###DynamicField**

Dynamic fields shown in the additional ITSM field screen of the agent interface.

## 2.4.34  Frontend::Agent::View::AgentITSMConfigItemZoom

ITSMConfigItem::Frontend::AgentITSMConfigItemZoom###DynamicField

Dynamic fields shown in the additional ITSM field screen of the agent interface.

## 2.4.35  Frontend::Agent::View::ConfigItemSearch

ITSMConfigItem::Frontend::AgentITSMConfigItemSearch###ExtendedSearchCondition

Allows extended search conditions in config item search of the agent interface. With this feature you can search e. g. config item name with this kind of conditions like "(key1*&&*key2)" or "(key1*||*key2)".

ITSMConfigItem::Frontend::AgentITSMConfigItemSearch###SearchPageShown

Number of config items to be displayed in each page of a search result in the agent interface.

ITSMConfigItem::Frontend::AgentITSMConfigItemSearch###SortBy::Default

Defines the default config item attribute for config item sorting of the config item search result of the agent interface.

ITSMConfigItem::Frontend::AgentITSMConfigItemSearch###Order::Default

Defines the default config item order in the config item search result of the agent interface. Up: oldest on top. Down: latest on top.

ITSMConfigItem::Frontend::AgentITSMConfigItemSearch###Defaults###Number

Defines the default shown config item search attribute for config item search screen.

ITSMConfigItem::Frontend::AgentITSMConfigItemSearch###Defaults###Name

Defines the default shown config item search attribute for config item search screen.

ITSMConfigItem::Frontend::AgentITSMConfigItemSearch###Defaults###DeploymentStateIDs

Defines the default shown config item search attribute for config item search screen.

ITSMConfigItem::Frontend::AgentITSMConfigItemSearch###Defaults###IncidentStateIDs

Defines the default shown config item search attribute for config item search screen.

ITSMConfigItem::Frontend::AgentITSMConfigItemSearch###Defaults###CustomerID

Defines the default shown config item search attribute for config item search screen.

**ITSMConfigItem::Frontend::AgentITSMConfigItemSearch###Defaults###CustomerUserLogin**

Defines the default shown config item search attribute for config item search screen.

**ITSMConfigItem::Frontend::AgentITSMConfigItemSearch###Defaults###ITSMConfigItemCreateTimePoint**

Default data to use on attribute for config item search screen. Example: "ITSMConfigItemCreateTime-PointFormat=year;ITSMConfigItemCreateTimePointStart=Last;ITSMConfigItemCreateTimePoint=2;".

**ITSMConfigItem::Frontend::AgentITSMConfigItemSearch###Defaults###ITSMConfigItemCreateTimeSlot**

Default data to use on attribute for config item search screen. Example: "ITSMConfigItemCreateTimeS-tartYear=2010;ITSMConfigItemCreateTimeStartMonth=10;ITSMConfigItemCreateTimeStartDay=4;ITSMConfigItemCreate

**ITSMConfigItem::Frontend::AgentITSMConfigItemSearch###Defaults###ITSMConfigItemChangeTimePoint**

Defines the default shown config item search attribute for config item search screen.

**ITSMConfigItem::Frontend::AgentITSMConfigItemSearch###Defaults###ITSMConfigItemChangeTimeSlot**

Defines the default shown config item search attribute for config item search screen.

**ITSMConfigItem::Frontend::AgentITSMConfigItemSearch###Defaults###DynamicField**

Defines the default shown config item search attribute for config item search screen. Example: "Key" must have the name of the Dynamic Field in this case 'X', "Content" must have the value of the Dynamic Field depending on the Dynamic Field type, Text: 'a text', Dropdown: '1', Date/Time: 'Search_DynamicField_XTimeSlotStartYear=1974; Search_DynamicField_XTimeSlotStartMonth=01; Search_DynamicField_XTimeSlotStartDay=26; Search_DynamicField_XTimeSlotStartHour=00; Search_DynamicField_XTimeSlotStartMinute=00; Search_DynamicField_XTimeSlotStartSecond=00; Search_DynamicField_XTimeSlotStopYear=2013; Search_DynamicField_XTimeSlotStopMonth=01; Search_DynamicField_XTimeSlotStopDay=26; Search_DynamicField_XTimeSlotStopHour=23; Search_DynamicField_XTimeSlotStopMinute=59; Search_DynamicField_XTimeSlotStopSecond=59;' and or 'Search_DynamicField_XTimePointFormat=week; Search_DynamicField_XTimePointStart=Before; Search_DynamicField_XTimePointValue=7;'.

## 2.4.36 Frontend::Agent::View::CustomerInformationCenter

**AgentCustomerInformationCenter::Backend###0060-CIC-ITSMConfigItemCustomerCompany**

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

## 2.4.37  Frontend::Agent::View::CustomerUserInformationCenter

**AgentCustomerUserInformationCenter::Backend###0060-CUIC-ITSMConfigItemCustomerUser**

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

## 2.4.38  Frontend::Agent::View::ITSMConfigItem

**ITSMConfigItem::Frontend::AgentITSMConfigItem###DefaultColumns**

Columns that can be filtered in the config item overview of the agent interface. Note: Only Config Item attributes and Dynamic Fields (DynamicField_NameX) are allowed.

**ITSMConfigItem::Frontend::AgentITSMConfigItem###SearchLimit**

Defines the search limit for the AgentITSMConfigItem screen.

**ITSMConfigItem::Frontend::AgentITSMConfigItem###DefaultCategory**

The default category which is shown, if none is selected.

**ITSMConfigItem::Frontend::AgentITSMConfigItem###ClassColumnsAvailable**

Defines the available columns of CIs in the config item overview depending on the CI class. Each entry must consist of a class name and an array of available fields for the corresponding class. Dynamic field entries have to honor the scheme DynamicField_FieldName.

**ITSMConfigItem::Frontend::AgentITSMConfigItem###ClassColumnsDefault**

Defines the default displayed columns of CIs in the config item overview depending on the CI class. Each entry must consist of a class name and an array of available fields for the corresponding class. Dynamic field entries have to honor the scheme DynamicField_FieldName.

## 2.4.39  Frontend::Agent::View::ITSMConfigItemBulk

**ITSMConfigItem::Frontend::AgentITSMConfigItemBulk###DeplState**

Sets the deployment state in the configuration item bulk screen of the agent interface.

**ITSMConfigItem::Frontend::AgentITSMConfigItemBulk###InciState**

Sets the incident state in the configuration item bulk screen of the agent interface.

## 2.4.40 Frontend::Agent::View::ITSMConfigItemDelete

ITSMConfigItem::Frontend::AgentITSMConfigItemDelete###Permission

Required privileges to delete config items.

## 2.4.41 Frontend::Agent::View::ITSMConfigItemEdit

ITSMConfigItem::Frontend::AgentITSMConfigItemEdit###RichTextWidth

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

ITSMConfigItem::Frontend::AgentITSMConfigItemEdit###RichTextHeight

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

## 2.4.42 Frontend::Agent::View::ITSMConfigItemHistory

ITSMConfigItem::Frontend::HistoryOrder

Shows the config item history (reverse ordered) in the agent interface.

## 2.4.43 Frontend::Agent::View::ITSMConfigItemSearch

ITSMConfigItem::Frontend::AgentITSMConfigItemSearch###SearchCSVData

Data used to export the search result in CSV format.

ITSMConfigItem::Frontend::AgentITSMConfigItemSearch###SearchCSVDynamicField

Dynamic Fields used to export the search result in CSV format.

ITSMConfigItem::Frontend::AgentITSMConfigItemSearch###Defaults###SearchPreviousVersions

Defines the default shown config item search attribute for config item search screen.

ITSMConfigItem::Frontend::AgentITSMConfigItemSearch###SearchLimit

Defines the search limit for the AgentITSMConfigItemSearch screen.

ITSMConfigItem::Frontend::AgentITSMConfigItemSearch###DynamicField

Dynamic fields shown in the config item search screen of the agent interface.

## 2.4.44 Frontend::Agent::View::Preferences

**PreferencesGroups###ConfigItemOverviewSmallPageShown**

Parameters for the pages (in which the configuration items are shown).

**PreferencesGroups###ConfigItemOverviewFilterSettings**

Parameters for the column filters of the small config item overview. Please note: setting 'Active' to 0 will only prevent agents from editing settings of this group in their personal preferences, but will still allow administrators to edit the settings of another user's behalf. Use 'PreferenceGroup' to control in which area these settings should be shown in the user interface.

## 2.4.45 Frontend::Base::DynamicFieldScreens

**DynamicFieldScreens###ITSMConfigurationManagement**

This configuration defines all possible screens to enable or disable dynamic fields.

**DefaultColumnsScreens###ITSMConfigurationManagement**

This configuration defines all possible screens to enable or disable default columns.

## 2.4.46 Frontend::Base::Loader

**Loader::Agent::CommonJS###100-ConfigurationManagement**

List of JS files to always be loaded for the agent interface.

## 2.4.47 Frontend::Base::NavBarModule

**Frontend::AdminModuleGroups###200-ITSMConfigurationManagement**

Defines available groups for the admin overview screen.

## 2.4.48 Frontend::Customer::ITSMConfigItemOverview

**ITSMConfigItem::Frontend::CustomerOverview###Small**

Defines an overview module to show the small view of a configuration item list.

## 2.4.49 Frontend::Customer::ModuleRegistration

**CustomerFrontend::Module###CustomerITSMConfigItem**

Frontend module registration for the customer interface.

**CustomerFrontend::Module###CustomerITSMConfigItemSearch**

Frontend module registration for the customer interface.

**CustomerFrontend::Module###CustomerITSMConfigItemZoom**

Frontend module registration for the customer interface.

**CustomerFrontend::Module###CustomerITSMConfigItemAttachment**

Frontend module registration for the customer interface.

## 2.4.50 Frontend::Customer::ModuleRegistration::Loader

**Loader::Module::CustomerITSMConfigItem###003-ITSMConfigItem**

Loader module registration for the customer interface.

**Loader::Module::CustomerITSMConfigItemSearch###003-ITSMConfigItem**

Loader module registration for the customer interface.

**Loader::Module::CustomerITSMConfigItemZoom###003-ITSMConfigItem**

Loader module registration for the customer interface.

## 2.4.51 Frontend::Customer::ModuleRegistration::MainMenu

**CustomerFrontend::Navigation###CustomerITSMConfigItem###003-ITSMConfigItem**

Main menu item registration.

## 2.4.52 Frontend::Customer::View::ConfigItemSearch

**ITSMConfigItem::Frontend::CustomerITSMConfigItemSearch###Permission**

Required permissions to use the ITSM configuration item search screen in the customer interface.

**ITSMConfigItem::Frontend::CustomerITSMConfigItemSearch###DeploymentState**

Includes deployment states in the config item search of the customer interface.

**ITSMConfigItem::Frontend::CustomerITSMConfigItemSearch###IncidentState**

Includes incident states in the config item search of the customer interface.

### ITSMConfigItem::Frontend::CustomerITSMConfigItemSearch###ExtendedSearchCondition

Allows extended search conditions in config item search of the customer interface. With this feature you can search e. g. config item name with this kind of conditions like "(key1*&&*key2)" or "(key1*||*key2)".

### ITSMConfigItem::Frontend::CustomerITSMConfigItemSearch###SearchPageShown

Number of config items to be displayed in each page of a search result in the customer interface.

### ITSMConfigItem::Frontend::CustomerITSMConfigItemSearch###SortBy::Default

Defines the default config item attribute for config item sorting of the config item search result of the customer interface.

### ITSMConfigItem::Frontend::CustomerITSMConfigItemSearch###Order::Default

Defines the default config item order in the config item search result of the customer interface. Up: oldest on top. Down: latest on top.

### ITSMConfigItem::Frontend::CustomerITSMConfigItemSearch###Defaults###Number

Defines the default shown config item search attribute for config item search screen.

### ITSMConfigItem::Frontend::CustomerITSMConfigItemSearch###Defaults###Name

Defines the default shown config item search attribute for config item search screen.

### ITSMConfigItem::Frontend::CustomerITSMConfigItemSearch###Defaults###DeploymentStateIDs

Defines the default shown config item search attribute for config item search screen.

### ITSMConfigItem::Frontend::CustomerITSMConfigItemSearch###Defaults###IncidentStateIDs

Defines the default shown config item search attribute for config item search screen.

### ITSMConfigItem::Frontend::CustomerITSMConfigItemSearch###Defaults###DynamicField

Defines the default shown config item search attribute for config item search screen. Example: "Key" must have the name of the Dynamic Field in this case 'X', "Content" must have the value of the Dynamic Field depending on the Dynamic Field type, Text: 'a text', Dropdown: '1', Date/Time: 'Search_DynamicField_XTimeSlotStartYear=1974; Search_DynamicField_XTimeSlotStartMonth=01; Search_DynamicField_XTimeSlotStartDay=26; Search_DynamicField_XTimeSlotStartHour=00; Search_DynamicField_XTimeSlotStartMinute=00; Search_DynamicField_XTimeSlotStartSecond=00; Search_DynamicField_XTimeSlotStopYear=2013; Search_DynamicField_XTimeSlotStopMonth=01; Search_DynamicField_XTimeSlotStopDay=26; Search_DynamicField_XTimeSlotStopHour=23; Search_DynamicField_XTimeSlotStopMinute=59; Search_DynamicField_XTimeSlotStopSecond=59;' and or 'Search_DynamicField_XTimePointFormat=week; Search_DynamicField_XTimePointStart=Before; Search_DynamicField_XTimePointValue=7;.

---

## 2.4.53 Frontend::Customer::View::ITSMConfigItem

**ITSMConfigItem::Frontend::CustomerITSMConfigItem###DefaultColumns**

Columns that can be filtered in the config item overview of the customer interface. Note: Only Config Item attributes and Dynamic Fields (DynamicField_NameX) are allowed.

**ITSMConfigItem::Frontend::CustomerITSMConfigItem###SearchLimit**

Defines the search limit for the CustomerITSMConfigItem screen.

**ITSMConfigItem::Frontend::CustomerITSMConfigItem###ClassColumnsAvailable**

Defines the available columns of CIs in the config item overview depending on the CI class. Each entry must consist of a class name and an array of available fields for the corresponding class. Dynamic field entries have to honor the scheme DynamicField_FieldName.

**ITSMConfigItem::Frontend::CustomerITSMConfigItem###ClassColumnsDefault**

Defines the default displayed columns of CIs in the config item overview depending on the CI class. Each entry must consist of a class name and an array of available fields for the corresponding class. Dynamic field entries have to honor the scheme DynamicField_FieldName.

## 2.4.54 Frontend::Customer::View::ITSMConfigItemOverview

**ITSMConfigItem::Frontend::CustomerITSMConfigItem###DynamicField**

Dynamic fields shown in the config item overview screen of the customer interface.

**ITSMConfigItem::Frontend::CustomerITSMConfigItemSearch###DynamicField**

Dynamic fields shown in the config item overview screen of the customer interface.

## 2.4.55 Frontend::Customer::View::ITSMConfigItemSearch

**ITSMConfigItem::Frontend::CustomerITSMConfigItemSearch###SearchLimit**

Defines the search limit for the CustomerITSMConfigItemSearch screen.

## 2.4.56 Frontend::Customer::View::ITSMConfigItemZoom

**ITSMConfigItem::Frontend::CustomerITSMConfigItemZoom###VersionsEnabled**

Customers can see historic CI versions.

ITSMConfigItem::Frontend::CustomerITSMConfigItemZoom###VersionsSelectable

Customers have the possibility to manually switch between historic CI versions.

ITSMConfigItem::Frontend::CustomerITSMConfigItemZoom###GeneralInfo

Which general information is shown in the header.

### 2.4.57 GenericInterface::Invoker

GenericInterface::Invoker::ConfigItemFetch::Classes

For every webservice (key) an array of classes (value) can be defined on which the import is restricted. For all chosen classes, or all existing classes the identifying attributes will have to be chosen in the invoker config.

### 2.4.58 GenericInterface::Invoker::ModuleRegistration

GenericInterface::Invoker::Module###Elasticsearch::ConfigItemManagement

GenericInterface module registration for the invoker layer.

GenericInterface::Invoker::Module###ConfigItem::ConfigItemFetch

GenericInterface module registration for the ConfigItemFetch invoker layer.

### 2.4.59 GenericInterface::Operation::ConfigItemCreate

GenericInterface::Operation::ConfigItemCreate###Permission

Defines Required permissions to create ITSM configuration items using the Generic Interface.

### 2.4.60 GenericInterface::Operation::ConfigItemDelete

GenericInterface::Operation::ConfigItemDelete###Permission

Defines Required permissions to delete ITSM configuration items using the Generic Interface.

### 2.4.61 GenericInterface::Operation::ConfigItemGet

GenericInterface::Operation::ConfigItemGet###Permission

Defines Required permissions to get ITSM configuration items using the Generic Interface.

## 2.4.62  GenericInterface::Operation::ConfigItemSearch

### GenericInterface::Operation::ConfigItemSearch###Permission

Defines Required permissions to search ITSM configuration items using the Generic Interface.

### GenericInterface::Operation::ConfigItemSearch###SearchLimit

Maximum number of config items to be displayed in the result of this operation.

### GenericInterface::Operation::ConfigItemSearch###SortBy::Default

Defines the default config item attribute for config item sorting of the config item search result of this operation.

### GenericInterface::Operation::ConfigItemSearch###Order::Default

Defines the default config item order in the config item search result of the this operation. Up: oldest on top. Down: latest on top.

## 2.4.63  GenericInterface::Operation::ConfigItemUpdate

### GenericInterface::Operation::ConfigItemUpdate###Permission

Defines Required permissions to update ITSM configuration items using the Generic Interface.

## 2.4.64  GenericInterface::Operation::ModuleRegistration

### GenericInterface::Operation::Module###ConfigItem::ConfigItemCreate

GenericInterface module registration for the operation layer.

### GenericInterface::Operation::Module###ConfigItem::ConfigItemGet

GenericInterface module registration for the operation layer.

### GenericInterface::Operation::Module###ConfigItem::ConfigItemUpdate

GenericInterface module registration for the operation layer.

### GenericInterface::Operation::Module###ConfigItem::ConfigItemSearch

GenericInterface module registration for the operation layer.

### GenericInterface::Operation::Module###ConfigItem::ConfigItemDelete

GenericInterface module registration for the operation layer.

## 2.5 Configuration Reference

**CMDB** Configuration Management Database

**CI** Configuration Item

About

## 3.1 Contact

Rother OSS GmbH
Email: hello@otobo.io
Web: https://otobo.io

## 3.2 Version

Author: Rother OSS GmbH / Version: rel-11.0 / Date of release: 2024-11-14

## 3.3 Terms and Conditions

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "COPYING".

Published by: Rother OSS GmbH, (https://otobo.io), Oberwalting 31, 94339 Leiblfing, Germany.