OTRS Extension ITSM-CIAttributeCollection

Installation and Usage

c.a.p.e. IT® GmbH

Version: 1.0.6 / 54

Title:	OTRS Extension ITSM-CIAttributeCollection
Topic:	Installation and Usage
Author:	c.a.p.e. IT® GmbH
Keywords:	OTRS, ITSM-CIAttributeCollection
Comments:	
Filename:	ITSM-CIAttributeCollection.odt
Version:	1.0.6 / 54
Set and Layout:	LibreOffice.org Version 4.2.x

Table of contents

1 General Remarks

1.1 Document information

1.1.1 Objectives

This document contains all the information to install and configure the OTRS Extension ITSM-CIAttributeCollection.

1.2 Changes

1.2.1 Change History

Version	Date	Changed Sections	Short description	Changed by
1.0.0	2013-04-13	all	initial document creation	Torsten Thau
1.0.1	2014-03-11	add 4.11; fix 4.2.1	added ticket reference	Mario Illinger
1.0.2	2014-04-23	add 4.12 and 4.13	added type reference and SLA reference	Anna Litvinova
1.0.3	2014-08-20	detailed 4.10	more precise description	Torsten Thau
1.0.4	2015-09-22	update 4.1	added new functions	Mario Illinger
1.0.5	2015-11-03	all	Document for OTRS 5.0 version	Andreas Hergert
1.0.6	2016-02-11	update 4.1	Added note for key	Torsten Thau

1.2.2 Authors

Lastname, Firstname	Organization	Role
Thau, Torsten	c.a.p.e. IT® GmbH	Head of Project Department
Illinger, Mario	c.a.p.e. IT® GmbH	SW-Ing.
Hergert, Andreas	c.a.p.e. IT® GmbH	

1.2.3 Document Owner

Organisation	Surname, first name	Address	Contacts
c.a.p.e. IT® GmbH		Schönherrstr. 8 09113 Chemnitz	info@cape-it.de +49 371 270 956 27

2 Installation

2.1 Requirements

To install and use the “OTRS Extension ITSM-CIAttributeCollection” your installation environment needs to fulfil the following requirements:

- Release 2.0.x: OTRS 3.1.x+ ITSMConfigurationManagement 3.1.1+
- Release 2.1.x: OTRS 3.2.x+ ITSMConfigurationManagement 3.2.1+
- Release 2.2.x: OTRS 3.2.x+ ITSMConfigurationManagement 3.2.1+
- Release 2.3.x: OTRS 3.3.x + ITSMConfigurationManagement 3.2.94+
- Release 4.1.x: OTRS 4.0.x + ITSMConfigurationManagement 4.0.0+
- Release 5.0.x: OTRS 5.0.x + ITSMConfigurationManagement 5.0.1+

2.2 Package Installation

You can install the package „ITSM-CIAttributeCollection“ via the Package Management in the OTRS GUI (Admin Area) or by using the command line interface package management tool.

3 Package Description

This package provides additional ITSM-CI-attributes for OTRS:ITSM, i.e. ITSMConfigurationManagement. Following you'll find a short documentation on the use and features of this extension.

3.1 Why not using VirtualFileSystem ?

Virtual file system proved to be not suitable, since it builds some sort of directory path out of object-class and id. Thus it would require to know the CIs version ID before it's actually created. If this could be solved somehow, copying a CI with an attachment would also cause a copy of the attachment itself, which is not desirable from our point of view.

3.2 Autocomplete function for attribute types

For some attribute types additional frontend modules were required in order to provide an autocomplete functionality. You may disable this if you prefere Drop-Down selection, but keep in mind, that the selection lists may become very long.

This functionality applies to Queue-, Service-, CIclass-Reference, Users, CIACCustomerCompany, CustomerUserCompany and Ticket-Reference. Its activation can be configured in SysConfig .

SysConfig Key	Description
Frontend::Module###AgentQueueSearch	registration for queue search frontend module
Frontend::Module###AgentUserSearch	registration for user search frontend module
Frontend::Module###AgentServiceSearch	registration for service search frontend module
Frontend::Module###AgentCustomerCompanySearch	registration for company search frontend module
Frontend::Module###AgentCustomerUserCompanySearch	registration for company user search frontend module
Frontend::Module###AgentCIClassSearch	registration for CI search frontend module
Frontend::Module###AgentSLASearch	registration for SLA search frontend module
ITSMCIAttributeCollection::Frontend::CommonSearchAutoComplete###-Active	dis-/enables auto complete input
ITSMCIAttributeCollection::Frontend::CommonSearchAutoComplete###-DynamicWidth	dis-/enables automatic width in autocomplete field
ITSMCIAttributeCollection::Frontend::CommonSearchAutoComplete###-MinQueryLength	defines minimum input lenght before auto complete search is started
ITSMCIAttributeCollection::Frontend::CommonSearchAutoComplete###-MaxResultsDisplayed	defines max. number of results displayed in auto complete search
ITSMCIAttributeCollection::Frontend::CommonSearchAutoComplete###-QueryDelay	defines the delay to start auto complete after the last key stroke
ITSMCIAttributeCollection::Frontend::CommonSearchAutoComplete###-TypeAhead	dis-/enables TypeAhead for the autocomplete feature

4 Configuration

4.1 Use of CI-Attribute "CIClassReference"

- behaves similar to customer user attribute but refers to another config item of some CI-class
- allows to automatically delete and create links to CIs of a referenced class
- eases the configuration of assembled config items, which consist (in parts) of other config items without annoying manual link creation

4.1.1 Additional Attribute Definition Parameters

- **ReferencedCIClassName** or ReferencedCIClassID: defines the referenced CI-class
- ***ReferencedCIClassLinkType***: defines the link type which should be used for automatic link creation and deletion, no link will be created if empty
- ***ReferencedCIClassLinkDirection***: set to "Reverse" if you want to create a reverse link for directed link types (will only be relevant if ReferencedCIClassLinkType is set)
- ***ReferencedCIClassReferenceAttributeKey***: defines the referenced attribute key which should be used as identifier during the import.
- ***SearchInputType***: defines the input type for the cmdb **search** mask. Can be 'AutoComplete' or 'Pattern'. If nothing or something wrong is set, 'AutoComplete' is used. 'Pattern' allows to search for several relevant configitems by name (* allowed as wildcard), by number (starts with '[Number]' followed by digest. * allowed as wildcard) and by id (starts with '[ID]' followed by digest). Multiple entries can be separated by '||'.
- **IMPORTANT NOTE:** If the CI will be edited using the edit-dialog all Links of the defined type will be deleted and set according to the CI-Attribute
- **IMPORTANT NOTES:**
  - The "Key" attribute has to be a UNIQUE name within the class definition!
  - Do NOT use non-alphanumeric characters in “Key” attribute”!

4.1.2 Example

```
{
    Key        => 'Location',
    Name       => 'Asset Location',
    Searchable => 1,
    Input => {
        Type                            => 'CIClassReference',
        ReferencedCIClassName           => 'Location',
        ReferencedCIClassLinkType       => 'Includes',
        ReferencedCIClassLinkDirection => 'Reverse',
        ReferencedCIClassReferenceAttributeKey => 'SerialNumber',
        SearchInputType                 => 'AutoComplete',
    },
},
```

4.2 Use of CI-Attribute "ServiceReference"

- behaves similar to customer user attribute but refers to a service
- allows to automatically delete and create links to Services

4.2.1 Additional Attribute Definition Parameters

- **ReferencedServiceLinkType**: defines the link type which should be used for automatic link creation and deletion, no link will be created if empty
- ***ReferencedServiceLinkDirection***: set to "Reverse" if you want to create a reverse link for directed link types (will only be relevant if ReferencedServiceLinkType is set)
- **IMPORTANT NOTE:** If the CI will be edited using the edit-dialog all Links of the defined type will be deleted and set according to the CI-Attribute
- **IMPORTANT NOTE:** If alle ServiceReferences in the CI-edit-dialog will be deleted (while Using Array-Elements) the Links will not be deleted
- **IMPORTANT NOTE**: The "Key" attribute has to be a UNIQUE name within the class definition!

4.2.2 Example

```
{
    Key        => 'AssignedService',
    Name       => 'Assigned Service',
    Searchable => 1,
    Input => {
        Type                            => 'ServiceReference',
        ReferencedServiceLinkType       => 'RelevantTo',
        ReferencedServiceLinkDirection  => 'Normal',
    },
},
```

4.3 Use of CI-Attribute "QueueReference"

- behaves similar to customer user attribute but refers to a queue

4.3.1 Additional Attribute Definition Parameters

- no additional parameters

4.3.2 Example

```
{
    Key        => 'AssignedService',
    Name       => 'Assigned Service',
    Searchable => 1,
    Input => {
        Type => 'QueueReference',

    },
},
```

4.4 Use of CI-Attribute "CIAttachment"

In contrast to standard OTRS CI-attachments, this attribute behaves much more like a common CI-attribute than the default CI-attachments (and had been there before standard attachments). Attachments are not limited to the maximum of 25 attachments, in fact you may define the number of allowed attachments and also make it a required attribute. Attachment files are bound to the version of a CI. Thus changing the attachment preserves it's previous version and causes the creation of a new CI-version.

- **ADVICE**: don't use this attribute in ITSMConfigItem Overview (KIX4OTRS or ITSMConfigItem::Frontend::AgentITSMConfigItem###ShowColumnsByClass)

4.4.1 Additional Attribute Definition Parameters

- no additional parameters, but CountMin and CountMax are required
- **IMPORTANT NOTE:** The attributes CountMin and CountMax must always be defined! Otherwise you won't be able to delete attachments from CIs. If you set CountMin => 1, it will be impossible to delete the last attachment (the last n attachments for CountMin => n).

4.4.2 Additional SysConfig-Options

- AttachmentStorage::StorageBackendModules
- AttachmentStorage::DefaultStorageBackendModule
- Frontend::Module###AgentAttachmentStorage
- AttachmentStorageFS::StorageDirectory
- ITSMCIAttributeCollection::AttachementMD5Check
- ITSMCIAttributeCollection::AttachementSizeCheck

4.4.3 Example

```
{
    Key=> 'Attachment',
    Name    => 'Attachment',
    Searchable => 1,
    Input => {
        Size  => 50,
        MaxLength => 50,
        Type  => 'CIAttachment',
    },
    CountMin => 0,
    CountMax => 1,
},
```

4.5 Use of CI-Attribute "User"

- behaves similar to customer user attribute but refers to a user/agent

4.5.1 Additional Attribute Definition Parameters

- no additional parameters

4.5.2 Example

```
{
    Key        => 'Responsible',
    Name       => 'Responsible',
    Searchable => 1,
    Input => {
        Type => 'User',

    },
},
```

4.6 Use of CI-Attribute "CustomerUserCompany"

- behaves similar to customer user attribute but refers to a CustomerCompany
- looks up in all Customer**User**Datas for CustomerCompany

4.6.1 Additional Attribute Definition Parameters

- no additional parameters

4.6.2 Example

```
{
    Key        => 'CustomerUserCompany',
    Name       => 'Customer User Company',
    Searchable => 1,
    Input => {
        Type => 'CustomerUserCompany',
    },
},
```

4.7 Use of CI-Attribute "CIACCustomerCompany"

- behaves similar to customer user attribute but refers to a CustomerCompany
- looks up in CustomerCompany-Backend
- attribute type renamed in r2.0.1 from "CustomerCompany" to "CIACCustomerCompany", since OTRS::ITSM 3.1.6 introduced a similar attribute type

4.7.1 Additional Attribute Definition Parameters

- no additional parameters

4.7.2 Example

```
{
    Key        => 'CustomerUserCompany',
    Name       => 'Customer Company',
    Searchable => 1,
    Input => {
        Type => 'CIACCustomerCompany',
    },
},
```

4.8 Use of CI-Attribute "EncryptedText"

- allows to deposite passwords or contents which should only be seen from certain OTRS-user-groups
- change and display of attribute only for the configured groups

4.8.1 Additional Attribute Definition Parameters

- **Group**: defines a Hash of OTRS-groups to which an user must belong in order to view this attribute (the user needs to be only in one of these groups)
- The user need “RW”-rights in one of these groups to watch and edit the encrypted text

4.8.2 Example

```
{
    Key        => 'Password',
    Name       => 'Password',
    Searchable => 1,
    Group => {
        admin => 1,
        SomeOtherGroupname => 1,
    },
    Input => {
        Type      => 'EncryptedText',
        Size      => 50,
        MaxLength => 100,
    },
},
```

4.9 Use of CI-Attribute "LinkText"

- allows to define a text that should be used in a configured link
- attribute URL must be defined and should contain the phrase "<VALUE>" which will be replaced with the given content
- this can be used to easily provide links to system monitoring tools or printer management interfaces
- if the link ist set using http/https the link will be absolute otherwise relative to the OTRS-instance
- **ADVICE**: don't use this attribute in ITSMConfigItem Overview (KIX4OTRS or ITSMConfigItem::Frontend::AgentITSMConfigItem###ShowColumnsByClass)

4.9.1 Additional Attribute Definition Parameters

- **URL**: defines HREF-destination for this value

4.9.2 Example

```
{
    Key        => 'TextLink',
    Name       => 'Text Link',
    Searchable => 1,
    Input => {
        Type      => 'TextLink',
        Size      => 50,
        MaxLength => 100,
        URL       => 'http://www.example.com/<VALUE>',
    },
},
```

4.10 Use of CI-Attribute "DynamicField"

- allows to use values of dropdown/single selection dynamic fields in config items

4.10.1 Additional Attribute Definition Parameters

- **Name**: defines the Dynamic Field Keyname
- **Translation**: enables value translation

4.10.2 Example

```
{
    Key        => 'Impact',
    Name       => 'Impact',
    Searchable => 1,
    Input => {
        Type        => 'DynamicField',
        Name        => 'TicketFreeText14',
        Translation => 1,
    },
},
```

4.11 Use of CI-Attribute "TicketReference"

- behaves similar to customer user attribute but refers to a ticket
- allows to automatically delete and create links to Tickets
- provides direct link to referenced ticket
- Searches are only possible to the Ticketnumber

4.11.1 Additional Attribute Definition Parameters

- **ReferencedTicketLinkType**: defines the link type which should be used for automatic link creation and deletion, no link will be created if empty
- ***ReferencedTicketLinkDirection***: set to "Reverse" if you want to create a reverse link for directed link types (will only be relevant if ReferencedCIClassLinkType is set)
- **IMPORTANT NOTE**: The "Key" attribute has to be a UNIQUE name within the class definition!
- **IMPORTANT NOTE:** If the CI will be edited using the edit-dialog all Links of the defined type will be deleted and set according to the CI-Attribute

4.11.2 Example

```
{
    Key        => 'RelatedTicket',
    Name       => 'Related Ticket',
    Searchable => 1,
    Input => {
        Type                         => 'TicketReference',
        ReferencedTicketLinkType     => 'RelevantTo',
        ReferencedTicketLinkDirection  => 'Normal',
    },
},
```

4.12 Use of CI-Attribute "TypeReference"

- behaves similar to customer user attribute but refers to a ticket type

4.12.1 Additional Attribute Definition Parameters

- no additional parameters

4.12.2 Example

```
{
    Key        => 'TicketType',
    Name       => 'Ticket Type',
    Searchable => 1,
    Input => {
        Type  => 'TypeReference',
    },
},
```

4.13 Use of CI-Attribute "SLAReference"

- behaves similar to service attribute but refers to an SLA and doesn't create links to SLAs

4.13.1 Additional Attribute Definition Parameters (optional)

- ***ReferencedServiceAttrKey***: defines dependency to an attribute of type “ServiceReference” form the current CI Class Definition. If configured, the value of the attribute with the given key will be used to filter the SLAs: Only SLAs that are configured for the given Service will be shown.

- ***ReferencedCustomerLogin***: defines dependency to an attribute of type “Customer” form the current CI Class Definition. If configured, the value of the attribute with the given key will be used to filter the SLAs: Only SLAs for the Services that are available for the given customer will be shown.
- ***ReferencedCustomerCompany***: defines dependency to an attribute of type “CIACCustomerCompany” form the current CI Class Definition. If configured, the value of the attribute with the given key will be used to filter the SLAs: Only SLAs available for the given CustomerID will be shown.

  !!! Requires KIXServiceKatalog to be installed!!!

  !!! If dependency to a CustomerUser (ReferencedCustomerLogin) is also confiugred, the ReferencedCustomerCompany will be ignored !!! So only one of these references can be used.
- **IMPORTANT NOTE**:

  If the referenced field does not exist or is empty, no SLAs will be found and only SLAs that are configured for DEFAULT services will be shown.
- **IMPORTANT NOTE**:

  If the referenced attribute is an array-attribut, SLAs for each referenced value will be found and only the intersection of these result sets will be shown.

  If SLAReference attribute is a sub-attribute of the referenced service- or customer(company) attribute, only the value of the superordinate attribute will be used.
- **!!! VERY !!! IMPORTANT NOTE**:

  For the fields of type “SLAReference” it is important to set the CountMin => 0 and CountDefault => 0 and to add these fields with the “+” button only AFTER all referenced fields (Service, CustomerUser, CustomerCompany) are filled with values. Otherwise the values of the referenced fields will be ignored!

4.13.2 Examples

- **SLAs for one (superordinate) Service and one CustomerUser**:

```
{
    Key => 'ResponsibleUser',
    Name => 'Responsible User',
    Searchable => 1,
    Input => {
        Type => 'Customer',
    },
},
{
    Key         => 'AssignedService',
    Name        => 'Assigned Service',
    Searchable => 1,
    Input => {
        Type                            => 'ServiceReference',
        ReferencedServiceLinkType       => 'RelevantTo',
        ReferencedServiceLinkDirection  => 'Normal',
    },
    CountMin => 0,
    CountMax => 10,
    CountDefault => 1,
    Sub => [
        {
            Key         => 'AssignedSLA',
            Name        => 'Assigned SLA',
            Searchable => 1,
            Input => {
                Type                        => 'SLAReference',
                ReferencedServiceAttrKey    => 'AssignedService',
                ReferencedCustomerLogin   => 'ResponsibleUser',
            },
            CountMin => 0,
            CountMax => 10,
           CountDefault => 0,
        },
    ],
},
```

- **SLAs for several (1 to 10) Services**:

```
{
    Key        => 'AssignedService',
    Name       => 'Assigned Service',
    Searchable => 1,
    Input => {
        Type                            => 'ServiceReference',
        ReferencedServiceLinkType       => 'RelevantTo',
        ReferencedServiceLinkDirection  => 'Normal',
    },
    CountMin => 0,
    CountMax => 10,
    CountDefault => 1,
},
{
            Key        => 'AssignedSLA',
            Name       => 'Assigned SLA',
            Searchable => 1,
            Input => {
                Type                        => 'SLAReference',
                ReferencedServiceAttrKey    => 'AssignedService',
            },
            CountMin => 0,
            CountMax => 10,
           CountDefault => 0,
},
```

- **SLAs for one CustomerCompany (only with KIXServiceGatalog!)**:

```
{
    Key        => 'CustomerCompany',
    Name       => 'CustomerCompany',
    Searchable => 1,
    Input => {
        Type => 'CIACCustomerCompany',
    },
},
{
       Key        => 'AssignedSLA',
       Name       => 'Assigned SLA',
       Searchable => 1,
       Input => {
                Type                        => 'SLAReference',
                ReferencedCustomerCompany  => 'CustomerCompany',
       },
       CountMin => 0,
       CountMax => 10,
       CountDefault => 0,
},
```