



OTRS ITSM Manual

Release 2024.6.1

OTRS AG

06.09.2024

1	Einführung	3
2	Configuration Management	5
2.1	Administrator-Interface	5
2.1.1	Benutzer, Gruppen & Rollen	5
2.1.1.1	Gruppen	6
2.1.2	Prozesse & Automatisierung	7
2.1.2.1	Prozessmanagement	7
2.1.2.2	Web-Services	25
2.1.3	Administration	27
2.1.3.1	General Catalog	27
2.1.3.2	Import und Export	28
2.1.3.3	Systemkonfiguration	34
2.1.4	CMDB-Einstellungen	37
2.1.4.1	Configuration Items	37
2.2	Agenten-Interface	43
2.2.1	Configuration Items	43
2.2.1.1	Configuration Item erstellen	43
2.2.1.2	Configuration Item-Liste	44
2.2.1.3	Configuration Item-Detailansicht	44
2.2.2	Kunden	46
2.2.3	Kundenbenutzer	47
2.3	Externes Interface	48

Das Copyright für dieses Werk liegt bei der OTRS AG (<https://otrs.com>), Zimmersmühlenweg 11, 61440 Oberursel, Deutschland.

Dieses Handbuch richtet sich an OTRS::ITSM Benutzer und Administratoren. Es vermittelt Informationen für die grundlegende Benutzung von OTRS::ITSM durch IT Service Manager, IT Servicemitarbeiter (Agenten) und Benutzer (Kunden). Informationen in Bezug auf die Installation, Konfiguration oder Administration von OTRS::ITSM werden nur erwähnt, wenn es Abweichungen zum OTRS Hauptprodukt gibt, oder wenn es um Funktionen geht, die in OTRS::ITSM nicht existieren.

Es wird erwartet, dass die IT beständig qualitativ hochwertige Dienstleistungen in einem zunehmend komplexer werdenden Umfeld erbringt. In diesem Zusammenhang ist ein leistungsfähiges und wirkungsvolles Vorfall- und Problem-Management erforderlich. IT Service-Management ist jedoch eine beinahe unlösbare Aufgabe, wenn keine aktuelle und konsistente Datenbank vorhanden ist, in der Informationen über den Status und die Konfiguration der IT-Infrastruktur gepflegt werden.

Die IT Infrastructure Library®, kurz ITIL®, ist eine Buchreihe, herausgegeben vom United Kingdom's Office of Government Commerce (OGC), die generisch Best-Practice-Ansätze für das Design, die Bereitstellung, den Betrieb und das Management von IT Dienstleistungen beschreibt. ITIL® fokussiert nicht auf die Technologie selbst, sondern auf die Dienstleistungen der IT und umfasst Informationen über Prozesse, Rollen, Verantwortlichkeiten, potenzielle Problembereiche/Lösungsansätze und Definitionen von Ausdrucksweisen.

ITIL hat sich in den vergangenen Jahre als de facto Standard durchgesetzt. Die Verbreitung in IT Organisationen hat zur Entwicklung einer kollektiven Wahrnehmung des IT Service Management und zur Schaffung einer einheitlichen Terminologie beigetragen. ITIL beschreibt jedoch lediglich, wer etwas tun sollte und was er tun sollte und was man während dieser Tätigkeit bedenken sollte. Um so viele Benutzergruppen wie möglich abzudecken, geht es nicht, oder meistens nur zu einem geringen Anteil, darum, zu beschreiben, was im Einzelfall getan werden muss. Aus diesem Grund gibt es keine direkt anwendbaren Informationen für spezielle Branchen, Firmen oder Hersteller.

Im Dezember 2005 wurde der ITIL®-basierte ISO/IEC 20000 Industriestandard veröffentlicht. IT-Organisationen können sich zur ISO/IEC 20000 Zertifizierung anmelden und ihre Konformität unter Beweis stellen.

Die ständige Weiterverbreitung verursachte eine Nachfrage nach IT Service Management Lösungen, die in der Lage waren, die ITIL®-basierten Prozesse abzubilden. Bis dahin gab es lediglich proprietäre Lösungen, die sich, aufgrund ihrer außerordentlichen Komplexität, nur große Unternehmungen leisten konnten, und die nur in großen Abteilungen wirkungsvoll einsetzbar waren.

Die Entwicklung von OTRS::ITSM wurde aufgrund des großen Erfolges des OTRS Frameworks gestartet, um die global anerkannten, öffentlichen ITIL®-Empfehlungen mit den Vorzügen von Open-Source Software zu vereinigen.

OTRS::ITSM 1.0 war die erste wirklich ITIL®-konforme Lösung für IT Service Management, die auf Open-Source basierte, aufgebaut auf Basis des stabilen OTRS mit seinen Zehntausend bekannten Installationen und seiner Benutzergemeinschaft.

OTRS::ITSM ist praxisorientiert. Die Entwicklung findet unter Teilnahme von ITIL®-Beratern und einigen Kunden aus OTRS-Gruppen statt.

Die Service-Desk und Ticket System Lösung OTRS ist die Basis für die ITIL-konforme IT Service Management Lösung OTRS::ITSM, mit Modulen für Incident Management, Problem Management, Service Level Management, Change- und Configuration Management und der integrierten CMDB.

Wie ITIL erhebt auch OTRS::ITSM nicht den Anspruch, eine *out-of-the-box* Lösung für alle Aufgaben und Fragestellungen im IT Service Management zu sein. Sie soll als flexible, stabile und leicht verständliche Informationsplattform dienen, die an die Anforderungen nahezu jedes Unternehmens angepasst werden kann.

Bitte entschuldigen Sie uns daher, dass wir Sie auf Folgendes aufmerksam machen: Der Einsatz eines ITIL-orientierten Tools wie OTRS::ITSM ist nur dann sinnvoll, wenn Prozesse, Personen und Produkte (IT-Services) wirklich ITIL-orientiert sind.

Ohne die durchdachte Anpassung der generischen ITIL-Prozesse an die Anforderungen des jeweiligen Geschäftsszenarios wird OTRS::ITSM keine spürbare Verbesserung der „Key Performance Indicators“ des IT Service Managements erreichen.

Sie sollten sich auch darüber im Klaren sein, dass erfolgreiche ITIL-Implementierungsprojekte in der Regel bis zu einem Jahr und länger dauern. Ihr Umfang und ihre Auswirkungen auf das Unternehmen sind nicht zu unterschätzen. Wir möchten jedoch darauf hinweisen, dass ein ordentlich implementiertes ITIL-orientiertes ITSM-Tool dazu beitragen kann, Zeit und Geld zu sparen, da die Prozessunterstützung des Tools den Prozess der organisatorischen Neuausrichtung unterstützt und beschleunigt.

Bemerkung: Die Implementierung von OTRS::ITSM basiert auf ITIL v4.

OTRS::ITSM unterstützt die folgenden Features und Prozesse, die in der Regel in der ersten Phase einer ITIL-Implementierung konzipiert werden:

- Vorfall-Verwaltung
- Problem-Management
- Service-Level-Management
- Configuration Management Database

Eine detailliertere Beschreibung der Nutzung und Anpassung des Systems finden Sie in den folgenden Abschnitten. Bitte beachten Sie, dass die einzelnen OTRS::ITSM-Pakete unabhängig voneinander installiert werden können und dass ihre Namen den jeweiligen ITIL-Themen entsprechen.

Bemerkung: Die ITSM Pakete werden vom *Customer Solution Team* in **OTRS** installiert. Bei *On-Premise*-Systemen kann der Kunde die Pakete mit dem Paketmanager installieren, wenn das *Customer Solution Team* die ausgewählten Pakete zum Repository hinzugefügt hat. Um ein Paket zu installieren, wenden Sie sich bitte an das *Customer Solution Team* über support@otrs.com oder im [OTRS Portal](#).

Configuration Management

Die Configuration Management Database (CMDB) ist keine Datenbank im technischen Sinne, sondern ein konzeptionelles IT-Modell, das für ein effizientes IT-Service-Management unerlässlich ist. Alle IT-Komponenten und Bestände werden in der CMDB verwaltet. Das Configuration Management geht über das Asset Management hinaus, das oft fälschlicherweise als Synonym verwendet wird, da es nicht nur Betriebsmittel aus finanzieller Sicht dokumentiert, sondern auch Informationen über die Beziehung zwischen Komponenten, Spezifikationen oder deren Standort erfasst. So kann der IT-Support schnell auf Informationen über die Interdependenz von IT-Services und den dafür notwendigen IT-Komponenten (z.B. Configuration Items) zugreifen.

Dieses Paket bietet ein Werkzeug zum Importieren und Exportieren von Configuration Items im CSV-Format.

2.1 Administrator-Interface

Dieses Kapitel beschreibt die neuen Funktionen, die im Administrator-Interface nach der Installation des Paketes verfügbar sind.

2.1.1 Benutzer, Gruppen & Rollen

Nach der Installation des Pakets wird dem System eine neue Gruppe hinzugefügt.

2.1.1.1 Gruppen

Nach der Installation des Pakets wird dem System eine neue Gruppe hinzugefügt. Die Ansicht zur Verwaltung der Gruppen ist im Modul *Gruppen* in der Gruppe *Benutzer, Gruppen & Rollen* verfügbar.

The screenshot shows the 'Group Management' interface. On the left, there is a sidebar with 'Actions' (Add Group), 'Filter for Groups' (Just start typing to filter...), and a 'Hint' box. The main area displays a table with 8 groups.

NAME	COMMENT	VALIDITY	CHANGED	CREATED
admin	Group of all administrators.	valid	11/11/2018 15:29 (Europe/Budapest)	11/11/2018 15:29 (Europe/Budapest)
itsm-configitem	Group for ITSM Configitem mask access in the agent interface.	valid	12/03/2018 12:11 (Europe/Budapest)	12/03/2018 12:11 (Europe/Budapest)
itsm-service	Group for ITSM Service mask access in the agent interface.	valid	11/30/2018 08:27 (Europe/Budapest)	11/29/2018 08:16 (Europe/Budapest)
stats	Group for statistics access.	valid	11/11/2018 15:29 (Europe/Budapest)	11/11/2018 15:29 (Europe/Budapest)
users	Group for default access.	valid	11/11/2018 15:29 (Europe/Budapest)	11/11/2018 15:29 (Europe/Budapest)

Hint:
The admin group is to get in the admin area and the stats group to get stats area.
Create new groups to handle access permissions for different groups of agent (e. g. purchasing department, support department, sales department, ...).
It's useful for ASP solutions.

Abb. 1: Gruppen-Verwaltung

Neue Gruppe

Nach der Installation des Pakets wird dem System die folgende Gruppe hinzugefügt:

itsm-configitem

Gruppe für den Zugriff auf die *Asset Management*-Ansichten des Agenten-Interfaces.

Bemerkung: Der primäre Administratorbenutzer (`root@localhost`) wird standardmäßig der Gruppe mit der Berechtigung *rw* hinzugefügt.

Siehe auch:

Um die richtigen Berechtigungen für andere Benutzer festzulegen, überprüfen Sie die folgenden Beziehungen:

- *Agenten Gruppen*
- *Kunden Gruppen*
- *Kundenbenutzer Gruppen*
- *Rollen Gruppen*

2.1.2 Prozesse & Automatisierung

Nach der Installation des Pakets werden zwei neue Skripte zum Skript-Task-Aktivitätselement der Prozessverwaltung hinzugefügt und einige neue Operationen zur generischen Schnittstelle hinzugefügt.

2.1.2.1 Prozessmanagement

Nach der Installation des Pakets werden fünf neue Module für Skript-Task-Aktivitäten und Sequenzfluss-Aktionen des Prozessmanagements hinzugefügt.

Prozessmodule

So sehen Sie die neuen Module:

1. Gehen Sie im Administrator-Interface zur Ansicht *Prozessverwaltung*.
2. Erstellen Sie einen neuen Prozess oder wählen Sie einen bestehenden Prozess aus, der eine Skript-Task-Aktivität enthält.
3. Klicken Sie auf den Eintrag *Aktivitäten* im Widget *Verfügbare Prozesselemente* in der linken Seitenleiste.
4. Erstellen Sie eine neue Skript-Task-Aktivität oder bearbeiten Sie eine bestehende.
5. Wählen Sie eines der neuen Skripte im Dropdown *Script*.
 - ITSMConfigItemDataPull
 - ITSMConfigItemDataPush
 - LinkWithITSMConfigItem
 - TicketLinkITSMConfigItem
 - TicketUpdateByLinkedCI
6. Klicken Sie auf die Schaltfläche *Speichern*, wenn die Schaltfläche *Konfigurieren* neben der Dropdown-Liste *Script* nicht sichtbar ist.
7. Klicken Sie auf die Schaltfläche *Konfigurieren*, um Parameter (Schlüssel-Werte-Paare) für das Skript hinzuzufügen.

ITSMConfigItemDataPull

Ein Modul zum Abrufen von Daten aus einem verknüpften ITSM Configuration Item.

Hauptsektion der Suchparameter

Die folgenden Parameter können für Einschränkungen verwendet werden:

- Klasse *
- Verwendungsstatus
- Vorfallstatus
- Verknüpfungstyp

Copy all specified attributes of a linked ITSM configuration item to the process ticket.

▼ Main search parameters for configuration items. Any results will be linked using the specified link type.

★ Class:

Deployment State:

Incident State:

Link Type:

Define the link type for found ITSMConfigItems.

▼ Additional configuration item matching conditions (e.g. key: "Number", value: "1,2"; key: "Name", value: "SomeName" or other XML attributes).

Key: Value:

▼ Desired behavior if more than one linked configuration item is found (matching all conditions).

★ Behavior

▼ Process ticket attributes to be updated by linked configuration item (e.g. key: "Title", value: "<OTRS_ITSMCI_HardDisk::Capacity::1>").

Key: Value:

Abb. 2: Prozessmanagement-Modul „ITSMConfigItemDataPull “

Zusätzlicher Abschnitt über die Bedingungen des Configuration Items

In diesem Abschnitt können Sie nach Configuration Items suchen.

Filter können mit Schlüssel-Werte-Paaren hinzugefügt werden. Es besteht eine logische UND-Beziehung zwischen den Filtern, wenn mehr als ein Filter hinzugefügt wird. Die logische ODER-Verknüpfung kann durch mehrere durch , getrennte Werte hinzugefügt werden.

Der Schlüssel `Limit` begrenzt die Anzahl der zurückgegebenen Configuration Items.

Abschnitt „Gewünschtes Verhalten “

Wenn im obigen Abschnitt mehrere Configuration Items zu finden sind, kann hier das gewünschte Verhalten definiert werden.

Mögliche Werte:

- Attribute aus dem zuerst gefundenen Configuration Item kopieren
- Kopieren Sie die Attribute aus dem zuletzt gefundenen Configuration Item
- Configuration Item ignorieren, nichts kopieren

Abschnitt „Attribute des Prozesstickets “

Mit diesem Modul können die Attribute des Prozesstickets aktualisiert werden. Der Schlüssel ist das Attribut des Prozesstickets. Der Wert kann ein vordefinierter Text, ein Attribut aus dem verknüpften Configuration Item in Form eines OTRS-Tags oder eine Verkettung von beidem sein. Das OTRS-Tag-Präfix `<OTRS_ITSMCI_*>` kann hier verwendet werden.

Beispiele:

Schlüssel	Wert
Priority	5 very high
DynamicField_Capacity	<OTRS_ITSMCI_HardDisk::Capacity::1>
Title	From: <OTRS_ITSMCI_Name>

Siehe auch:

Schauen Sie in der *ITSMConfigItemDataPull* und der *ConfigItemSearch()* API-Referenz.

ITSMConfigItemDataPush

Ein Modul zum Einfügen von Daten in ein verknüpftes ITSM Configuration Item.

Copy all specified attributes to matching linked ITSM configuration items.

▼ Main search parameters for configuration items. Any results will be linked using the specified link type.

★ Class:

Deployment State:

Incident State:

Link Type:

Define the link type for found ITSMConfigItems.

▼ Additional configuration items matching conditions (e.g. key: "Number", value: "1,2"; key: "Name", value: "SomeName" or other XML attributes). ⊕

Key: Value:

▼ Linked configuration items attributes to be updated (e.g. key: "HardDisk::Capacity::1", value: "<OTRS_TICKET_DynamicField_HDCapacity>"). ⊕

Key: Value:

Abb. 3: Prozessmanagement-Modul „ITSMConfigItemDataPush“

Hauptsektion der Suchparameter

Die folgenden Parameter können für Einschränkungen verwendet werden:

- Klasse *
- Verwendungsstatus
- Vorfallstatus
- Verknüpfungstyp

Zusätzlicher Abschnitt über die Bedingungen des Configuration Items

In diesem Abschnitt können Sie nach Configuration Items suchen.

Filter können mit Schlüssel-Werte-Paaren hinzugefügt werden. Es besteht eine logische UND-Beziehung zwischen den Filtern, wenn mehr als ein Filter hinzugefügt wird. Die logische ODER-Verknüpfung kann durch mehrere durch , getrennte Werte hinzugefügt werden.

Der Schlüssel `Limit` begrenzt die Anzahl der zurückgegebenen Configuration Items.

Abschnitt „Verknüpfte Configuration Item-Attribute“

Hier können die zu aktualisierenden Attribute des verknüpften Configuration Items festgelegt werden. Der Schlüssel ist das Attribut des verknüpften Configuration Items. Der Wert kann ein vordefinierter Text, ein Attribut aus dem Prozessticket in Form eines OTRS-Tags oder eine Verkettung von beidem sein. Das OTRS-Tag-Präfix `<OTRS_TICKET_*>` kann hier verwendet werden.

Beispiele:

Schlüssel	Wert
ConfigItemCreateTimeNewerThan	2021-10-20 12:23:34
HardDisk::Capacity::1	<OTRS_TICKET_DynamicField_HDCapacity>
Name	Process: <OTRS_TICKET_DynamicField_ProcessManagementProcessID

Siehe auch:

Schauen Sie in der [ITSMConfigItemDataPull](#) und der [ConfigItemSearch\(\)](#) API-Referenz.

LinkWithITSMConfigItem

Ein Modul zur Verknüpfung von ITSM Configuration Items.

Search for one or more configuration items and link all matches to the process ticket.

▼ Main search parameters for configuration items. Any results will be linked using the specified link type.

★ Class:

Deployment State:

Incident State:

Link Type:

Define the link type for found ITSMConfigItems.

▼ Additional configuration item matching conditions (e.g. key: "Number", value: "1,2"; key: "Name", value: "SomeName" or other XML attributes). ⊕

Key: Value:

Key: Value:

Key: Value:

Abb. 4: Prozessmanagement-Modul „LinkWithITSMConfigItem“

Hauptsektion der Suchparameter

Die folgenden Parameter können für Einschränkungen verwendet werden:

- Klasse *
- Verwendungsstatus
- Vorfallstatus
- Verknüpfungstyp

Zusätzlicher Abschnitt über die Bedingungen des Configuration Items

In diesem Abschnitt können Sie nach Configuration Items suchen.

Filter können mit Schlüssel-Werte-Paaren hinzugefügt werden. Es besteht eine logische UND-Beziehung zwischen den Filtern, wenn mehr als ein Filter hinzugefügt wird. Die logische ODER-Verknüpfung kann durch mehrere durch , getrennte Werte hinzugefügt werden.

Der Schlüssel `Limit` begrenzt die Anzahl der zurückgegebenen Configuration Items.

Siehe auch:

Schauen Sie in der [LinkWithITSMConfigItem](#) und der [ConfigItemSearch\(\)](#) API-Referenz.

TicketLinkITSMConfigItem

Ein Modul zur Durchführung einer erweiterten Configuration Item-Suche und zur Verknüpfung von Suchergebnissen mit Tickets.

Search for one or more configuration items and link all matches to the ticket.

▼ Main search parameters for configuration items. Any results will be linked using the specified link type.

★ Class:

Deployment State:

Incident State:

Link Type:

Define the link type for found ITSMConfigItems.

▼ Additional attributes for configuration item search (e.g. OrderBy, OrderByDirection and XML definition attributes). ⊕

Key: Value:

Abb. 5: Prozessmanagement-Modul „TicketLinkITSMConfigItem“

Hauptsektion der Suchparameter

Die folgenden Parameter können für Einschränkungen verwendet werden:

- Klasse *
- Verwendungsstatus
- Vorfallstatus
- Verknüpfungstyp

Zusätzlicher Abschnitt über die Bedingungen des Configuration Items

In diesem Abschnitt können Sie nach Configuration Items suchen.

Filter können mit Schlüssel-Werte-Paaren hinzugefügt werden. Es besteht eine logische UND-Beziehung zwischen den Filtern, wenn mehr als ein Filter hinzugefügt wird. Die logische ODER-Verknüpfung kann durch mehrere durch , getrennte Werte hinzugefügt werden.

Sie können die Suchergebnisse sortieren, wenn Sie den Schlüssel `OrderBy` und das Attribut eines Configuration Items als Wert angeben.

Sie können die Sortierreihenfolge mit dem Schlüssel `OrderByDirection` und den Werten `Up` oder `Down` beeinflussen.

Wenn keine Sortierung angegeben wird, erfolgt die Sortierung automatisch in absteigender Reihenfolge anhand der ID der Configuration Items.

Siehe auch:

Schauen Sie in der [TicketLinkITSMConfigItem](#) und der [ConfigItemSearch\(\)](#) API-Referenz.

TicketUpdateByLinkedCI

Ein Modul zum Festlegen von Ticket-Attributen basierend auf einem verknüpften Configuration Item.

Copy all specified attributes of a linked configuration item to the ticket.

▼ Restrictions for linked configuration items. Only configuration items linked with the specified link type and matching other restrictions will be considered.

Class:

Deployment State:

Incident State:

Link Type:

Define the link type for found ITSMConfigItems.

▼ Desired behavior if more than one linked configuration item is found (matching all conditions).

★ Behavior

▼ Mapping of configuration item attributes to ticket attributes (e.g. key: "HardDisk::Capacity", value: "DynamicField_HardDiskSize" or key: "Name", value: "Title").

Key: Value:

Abb. 6: Prozessmanagement-Modul „TicketUpdateByLinkedCI“

Hauptsektion der Suchparameter

Die folgenden Parameter können für Einschränkungen verwendet werden:

- Klasse
- Verwendungsstatus
- Vorfallstatus
- Verknüpfungstyp

Abschnitt „Gewünschtes Verhalten“

Wenn im obigen Abschnitt mehrere Configuration Items zu finden sind, kann hier das gewünschte Verhalten definiert werden.

Mögliche Werte:

- Attribute aus dem zuerst gefundenen Configuration Item kopieren
- Kopieren Sie die Attribute aus dem zuletzt gefundenen Configuration Item
- Configuration Item ignorieren, nichts kopieren

Zuordnung eines Configuration Items zu einem Ticket-Bereich

Mit diesem Modul können die Attribute des Configuration Items auf Ticket-Attribute abgebildet werden. Der Schlüssel ist das Attribut des Configuration Items. Der Wert ist das Attribut des Tickets.

Beispiele:

Schlüssel	Wert
HardDisk::Capacity	DynamicField_HDCapacity
Name	Title

Siehe auch:

Schauen Sie in der [TicketUpdateByLinkedCI](#) und der [ConfigItemSearch\(\)](#) API-Referenz.

API-Referenz

Diese API-Referenzen sind nicht online verfügbar, aber sie sind in diesem Handbuch enthalten.

ITSMConfigItemDataPull API

```

NAME

Kernel::System::ProcessManagement::Modules::ITSMConfigItemDataPull - A module to
↳ fetch data from a linked ITSM configuration item.

DESCRIPTION

All ITSMConfigItemDataPull functions.

PUBLIC INTERFACE

new()

Don't use the constructor directly, use the ObjectManager instead:

    my $ITSMConfigItemDataPullObject = $Kernel::OM->Get (
↳ 'Kernel::System::ProcessManagement::Modules::ITSMConfigItemDataPull');

Run()

Run Data

my $Success = $ITSMConfigItemDataPullObject->Run (
    UserID                => 123,
    Ticket                 => \%Ticket, # required
    ProcessEntityID       => 'P123',
    ActivityEntityID      => 'A123',
    SequenceFlowEntityID  => 'T123',
    SequenceFlowActionEntityID => 'TA123',
    Config => {

        ConfigITSMConfigItemSearch => {
            ClassIDs    => [9, 8, 7, 6],           # (optional)
            DeplStateIDs => [1, 2, 3, 4],         # (optional)
            InciStateIDs => [1, 2, 3, 4],         # (optional)
            LinkType    => 'RelevantTo::Source',  # (optional)
        },

        ConfigSearchKeyValueList => {
            Name          => 'The Name',          # (optional)

            # configuration items with created time after ...
            ConfigItemCreateTimeNewerDate => '2006-01-09 00:00:01', # (optional)
            # configuration items with created time before then ....
            ConfigItemCreateTimeOlderDate => '2006-01-19 23:59:59', # (optional)

            # configuration items with changed time after ...
            ConfigItemChangeTimeNewerDate => '2006-01-09 00:00:01', # (optional)
            # configuration items with changed time before then ....

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

        ConfigItemChangeTimeOlderDate => '2006-01-19 23:59:59', # (optional)

        # XML attributes (defined by class)
        'ElementA::ElementB' => '%contentA%',
        'ElementA::ElementB' => '%contentC%,%contentD%,%contentE%',
    }.

    UserID => 123,      # optional,to override the UserID from the logged user

    ConfigDropdown => {
        Behavior => 'NoCopy',      # 'NoCopy', 'CopyFirstLinked' or
↪ 'CopyLastLinked' only
    },

    UserID => 123,      # optional,to override the UserID from the logged user

    # Value set:
    # * Key is the attribute of the linked ITSM configuration item where the
↪data will be set,
    # * Value is the value is the value to be set, supporting smart tags <OTRS_
↪ITSMCI_*> from the resulting linked
    # configuration item after match and behavior filters
    #
    # Example:
    # * To set process ticket title to be exactly the linked configuration item
↪first element of field someDefinitionField::Sub
    # Title => '<OTRS_ITSMCI_someDefinitionField::Sub::1>',
    # where 'Sub' is a sub field of field 'someDefinitionField'
    # * To set process ticket title to be exactly the linked configuration item
↪second element of field someDefinitionField
    # Title => '<OTRS_ITSMCI_someDefinitionField2::2>',
    # * To set the process ticket dynamic field NameX to an static value (not
↪real pull):
    # DynamicField_NameX => 'someValue',
    }
);

- `Ticket` contains the result of TicketGet including DynamicFields.

```

ITSMConfigItemDataPush API

```

NAME

Kernel::System::ProcessManagement::Modules::ITSMConfigItemDataPush - A module to
↪insert data to a linked ITSM configuration items.

DESCRIPTION

All ITSMConfigItemDataPush functions.

PUBLIC INTERFACE

new()

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

Don't use the constructor directly, use the ObjectManager instead:

    my $ITSMConfigItemDataPushObject = $Kernel::OM->Get (
↳ 'Kernel::System::ProcessManagement::Modules::ITSMConfigItemDataPush');

Run()

Run Data

my $Success = $ITSMConfigItemDataPullObject->Run(
    UserID                => 123,
    Ticket                => \%Ticket, # required
    ProcessEntityID      => 'P123',
    ActivityEntityID     => 'A123',
    SequenceFlowEntityID => 'T123',
    SequenceFlowActionEntityID => 'TA123',
    Config => {

        ConfigITSMConfigItemSearch => {
            ClassIDs      => [9, 8, 7, 6],          # (optional)
            DeplStateIDs => [1, 2, 3, 4],          # (optional)
            InciStateIDs => [1, 2, 3, 4],          # (optional)
            LinkType      => 'RelevantTo::Source', # (optional)
        },

        ConfigSearchKeyValueList => {
            Name           => 'The Name',          # (optional)

            # configuration items with created time after ...
            ConfigItemCreateTimeNewerDate => '2006-01-09 00:00:01', # (optional)
            # configuration items with created time before then ....
            ConfigItemCreateTimeOlderDate => '2006-01-19 23:59:59', # (optional)

            # configuration items with changed time after ...
            ConfigItemChangeTimeNewerDate => '2006-01-09 00:00:01', # (optional)
            # configuration items with changed time before then ....
            ConfigItemChangeTimeOlderDate => '2006-01-19 23:59:59', # (optional)

            # XML attributes (defined by class)
            'ElementA::ElementB' => '%contentA%',
            'ElementA::ElementB' => '%contentC%,%contentD%,%contentE%',
        },

        UserID => 123, # optional, to override the UserID from the logged user

        ConfigDropdown => {
            Behavior => 'NoCopy', # 'NoCopy', 'CopyFirstLinked' or
↳ 'CopyLastLinked' only
        },

        UserID => 123, # optional, to override the UserID from the logged user

        # Value set:
        # * Key is the attribute of the linked ITSM configuration items where the
↳ data will be pushed,
        # * Value is the value is the value to be set, supporting smart tags <OTRS_

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

↪TICKET_*> from the current process ticket e.g.
#
# Example:
# * To set linked ITSM configuration items first element of ↪
↪someDefinitionField::Sub to be exactly the process ticket QueueID:
# 'someDefinitionField::Sub::1' => '<OTRS_Ticket_QueueID>',
# Where 'Sub' is a sub field of 'someDefinitionField' field
# * To set linked ITSM configuration items second element of ↪
↪someDefinitionField2 to be the concatenation of
# some text and the content of the process ticket dynamic field ↪
↪ExternalField2:
# 'someDefinitionField2::2' => 'Some text <OTRS_Ticket_DynamicField_
↪ExternalField2>',
# * To set linked ITSM configuration items last element of ↪
↪someDefinitionField::Sub to be an static text:
# 'someDefinitionField::Sub' => 'Some text',
}
);
- `Ticket` contains the result of TicketGet including DynamicFields.

```

LinkWithITSMConfigItem API

```

NAME
Kernel::System::ProcessManagement::Modules::LinkWithITSMConfigItem - A module to link ↪
↪ITSM configuration items.

DESCRIPTION
All LinkWithITSMConfigItem functions.

PUBLIC INTERFACE
new()
Don't use the constructor directly, use the ObjectManager instead:
    my $LinkWithITSMConfigItemObject = $Kernel::OM->Get(
↪'Kernel::System::ProcessManagement::Modules::LinkWithITSMConfigItem');
Run()
Run Data
my $Success = $LinkWithITSMConfigItem->Run(
    UserID           => 123,
    Ticket           => \%Ticket, # required
    ProcessEntityID => 'P123',
    ActivityEntityID => 'A123',
    SequenceFlowEntityID => 'T123',
    SequenceFlowActionEntityID => 'TA123',
    Config => {

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

ConfigITSMConfigItemSearch => {
    ClassIDs      => [9, 8, 7, 6],          # (optional)
    DeplStateIDs => [1, 2, 3, 4],          # (optional)
    InciStateIDs => [1, 2, 3, 4],          # (optional)
    LinkType      => 'RelevantTo::Source', # (optional)
},

ConfigSearchKeyValueList => {
    Number      => 'The ConfigItem Number', # (optional)
    Name        => 'The Name',              # (optional)

    # configuration items with created time after ...
    ConfigItemCreateTimeNewerDate => '2006-01-09 00:00:01', # (optional)
    # configuration items with created time before then ...
    ConfigItemCreateTimeOlderDate => '2006-01-19 23:59:59', # (optional)

    # configuration items with changed time after ...
    ConfigItemChangeTimeNewerDate => '2006-01-09 00:00:01', # (optional)
    # configuration items with changed time before then ...
    ConfigItemChangeTimeOlderDate => '2006-01-19 23:59:59', # (optional)

    # XML attributes (defined by class)
    'ElementA::ElementB' => '%contentA%',
    'ElementA::ElementB' => '%contentC%,%contentD%,%contentE%',
}.

    UserID => 123,      # optional, to override the UserID from the logged user
}
);

- `Ticket` contains the result of TicketGet including DynamicFields.

```

TicketLinkITSMConfigItem API

```

NAME

Kernel::System::ProcessManagement::Modules::TicketLinkITSMConfigItem - A module to
↳ perform an extended configuration item search and link search results to ticket.

DESCRIPTION

All TicketLinkITSMConfigItem functions.

PUBLIC INTERFACE

new()

Don't use the constructor directly, use the ObjectManager instead:

    my $TicketLinkITSMConfigItemObject = $Kernel::OM->Get(
↳ 'Kernel::System::ProcessManagement::Modules::TicketLinkITSMConfigItem');

Run()

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

Run Data

my $Success = $TicketLinkITSMConfigItemObject->Run(
    UserID          => 123,
    Ticket          => \%Ticket,      # required
    ProcessEntityID => 'P123',
    ActivityEntityID => 'A123',
    SequenceFlowEntityID => 'T123',
    SequenceFlowActionEntityID => 'TA123',
    Config => {
        UserID          => 123,      # optional, to override the
↪UserID from the logged user
    }
);

- `Ticket` contains the result of TicketGet including DynamicFields.
- `Config` is the Config Hash stored in a Process::SequenceFlowAction's Config key.

```

TicketUpdateByLinkedCI API

```

NAME

Kernel::System::ProcessManagement::Modules::TicketUpdateByLinkedCI - A module to set
↪ticket attributes based on a linked configuration item.

DESCRIPTION

All TicketUpdateByLinkedCI functions.

PUBLIC INTERFACE

new()

Don't use the constructor directly, use the ObjectManager instead:

my $TicketUpdateByLinkedCIObject = $Kernel::OM->Get(
↪'Kernel::System::ProcessManagement::Modules::TicketUpdateByLinkedCI');

Run()

Run Data

my $Success = $TicketUpdateByLinkedCIObject->Run(
    UserID          => 123,
    Ticket          => \%Ticket,      # required
    ProcessEntityID => 'P123',
    ActivityEntityID => 'A123',
    SequenceFlowEntityID => 'T123',
    SequenceFlowActionEntityID => 'TA123',
    Config => {
        ConfigITSMConfigItemSearch => {
            ClassID          => 123,      # optional
            DeplStateIDs     => [123],    # optional
            InciStateIDs     => [123],    # optional

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

        LinkType                => 'someType::someDirection', # optional
    },
    ConfigDropdown => {
        Behavior                => 'NoCopy',                  # 'CopyFirstLinked
↪', 'CopyLastLinked'
    },
        'someDefinitionField::Sub' => 'DynamicField_someName', # example optional
        'someDefinitionField2'    => 'someTicketAttribute',    # example optional
        UserID                    => 123,                      # optional, to
↪override the UserID from the logged user
    }
);

```

- `Ticket` contains the result of TicketGet including DynamicFields.
- `Config` is the Config Hash stored in a Process::SequenceFlowAction's Config key.

ConfigItemCreate () API

perform ConfigItemCreate Operation. This will return the created config item number.

```

my $Result = $OperationObject->Run(
    Data => {
        UserLogin                => 'some agent login',        # UserLogin or
↪AccessToken is required
        AccessToken              => 'eyJhbGciOiJIUzI1NiJ9[...]',

        Password                => 'some password',            # if UserLogin is sent then
                                                                    # Password is required

        ConfigItem => {
            Number                => '111',                    # optional
            Class                 => 'Configuration Item Class',
            Name                  => 'The Name',
            DeplState             => 'deployment state',
            InciState             => 'incident state',
            CIXMLData             => $ArrayHashRef,             # it depends on the
↪Configuration Item class and definition

            Attachment => [
                {
                    Content        => 'content'                # base64 encoded
                    ContentType    => 'some content type'
                    Filename        => 'some fine name'
                },
                # ...
            ],
            #or
            #Attachment => {
            #    Content            => 'content'
            #    ContentType        => 'some content type'
            #    Filename            => 'some fine name'
            #},
        },
    },
);

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

$Result = {
    Success          => 1,                # 0 or 1
    ErrorMessage     => '',              # in case of error
    Data             => {                # result data payload after Operation
        ConfigItemID => 123,             # Configuration Item ID number in
↳OTRS::ITSM (Service desk system)
        Number       => 2324454323322   # Configuration Item Number in
↳OTRS::ITSM (Service desk system)
        Error => {                      # should not return errors
            ErrorCode => 'ConfigItemCreate.ErrorCode'
            ErrorMessage => 'Error Description'
        },
    },
};

```

ConfigItemDelete() API

perform ConfigItemDelete Operation. This function is able to return one or more ConfigItem entries in one call.

```

my $Result = $OperationObject->Run(
    Data => {
        UserLogin          => 'some agent login',          # UserLogin or
↳CustomerUserLogin or AccessToken is
        CustomerUserLogin => 'some customer login',
        AccessToken       => 'eyJhbGciOiJIUzI1NiJ9[...]',
        Password         => 'some password',              # if UserLogin or
↳customerUserLogin is sent then
        ConfigItemID     => '32,33',                      # Password is required
↳coma separated IDs or an Array
    },
);

$Result = {
    Success          => 1,                # 0 or 1
    ErrorMessage     => '',              # in case of error
    Data             => {                # result data payload after Operation
        ConfigItemID => [123, 456],      # Configuration Item IDs number in
↳OTRS::ITSM (Service desk system)
        Error => {                      # should not return errors
            ErrorCode => 'ConfigItemDelete.ErrorCode'
            ErrorMessage => 'Error Description'
        },
    },
};

```


ConfigItemGet () API

perform ConfigItemGet Operation. This function is able to return one or more ConfigItem entries in one call.

```

my $Result = $OperationObject->Run(
    Data => {
        UserLogin          => 'some agent login',          # UserLogin or
↳Access token is
        AccessToken       => 'eyJhbGciOiJIUzI1NiJ9[...]', # required
        Password          => 'some password',            # if UserLogin is sent
↳then Password is required
        ConfigItemID     => '32,33',                      # required, could be
↳coma separated IDs or an Array
        Attachments       => 1,                          # Optional, 1 as default.
↳ If it's set with the value 1,
                                                    # attachments for
↳articles will be included on ConfigItem data
    },
);

$Result = {
    Success      => 1,          # 0 or 1
    ErrorMessage => '',       # In case of an error
    Data        => {
        ConfigItem => [
            {
                Number          => '20101027000001',
                ConfigItemID   => 123,
                Name            => 'some name',
                Class           => 'some class',
                VersionID      => 123,
                LastVersionID  => 123,
                DefinitionID   => 123,
                InciState      => 'some incident state',
                InciStateType  => 'some incident state type',
                DeplState      => 'some deployment state',
                DeplStateType  => 'some deployment state type',
                CurInciState   => 'some incident state',
                CurInciStateType => 'some incident state type',
                CurDeplState   => 'some deployment state',
                CurDeplStateType => 'some deployment state type',
                CreateTime     => '2010-10-27 20:15:00'
                CreateBy      => 123,
                CIXMLData     => $XMLDataHashRef,

                Attachment => [
                    {
                        Content          => "xxxx",          # actual attachment
↳contents, base64 encoded
                        ContentType     => "application/pdf",
                        Filename        => "StdAttachment-Test1.pdf",
                        Filesize       => "4.6 KBytes",
                        Preferences     => $PreferencesHashRef,
                    },
                ],
            }
        ]
    }
}

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

        # . . .
    },
    ],
},
{
    # . . .
},
],
},
};

```

ConfigItemSearch() API

ConfigItemSearch()

return a configuration item list as an array reference

```

my $ConfigItemIDs = $ConfigItemObject->ConfigItemSearch(
    Number      => 'The ConfigItem Number', # (optional)
    ClassIDs    => [9, 8, 7, 6],           # (optional)
    DeplStateIDs => [1, 2, 3, 4],         # (optional)
    InciStateIDs => [1, 2, 3, 4],         # (optional)
    CreateBy    => [1, 2, 3],            # (optional)
    ChangeBy    => [3, 2, 1],            # (optional)

    # configuration items with created time after ...
    ConfigItemCreateTimeNewerDate => '2006-01-09 00:00:01', # (optional)
    # configuration items with created time before then ....
    ConfigItemCreateTimeOlderDate => '2006-01-19 23:59:59', # (optional)

    # configuration items with changed time after ...
    ConfigItemChangeTimeNewerDate => '2006-01-09 00:00:01', # (optional)
    # configuration items with changed time before then ....
    ConfigItemChangeTimeOlderDate => '2006-01-19 23:59:59', # (optional)

    OrderBy => [ 'ConfigItemID', 'Number' ], # (optional)
    # default: [ 'ConfigItemID' ]
    # (ConfigItemID, Number, ClassID, DeplStateID, InciStateID,
    # CreateTime, CreateBy, ChangeTime, ChangeBy)

    # Additional information for OrderBy:
    # The OrderByDirection can be specified for each OrderBy attribute.
    # The pairing is made by the array indices.

    OrderByDirection => [ 'Down', 'Up' ], # (optional)
    # default: [ 'Down' ]
    # (Down | Up)

    Limit      => 122, # (optional)
    UsingWildcards => 0, # (optional) default 1
);

```

ConfigItemSearchExtended()

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

return a configuration item list as an array reference

my $ConfigItemIDs = $ConfigItemObject->ConfigItemSearchExtended(
    Number      => 'The ConfigItem Number', # (optional)
    Name        => 'The Name',              # (optional)
    ClassIDs    => [9, 8, 7, 6],            # (optional)
    DeplStateIDs => [1, 2, 3, 4],           # (optional)
    InciStateIDs => [1, 2, 3, 4],           # (optional)

    # configuration items with created time after ...
    ConfigItemCreateTimeNewerDate => '2006-01-09 00:00:01', # (optional)
    # configuration items with created time before then ....
    ConfigItemCreateTimeOlderDate => '2006-01-19 23:59:59', # (optional)

    # configuration items with changed time after ...
    ConfigItemChangeTimeNewerDate => '2006-01-09 00:00:01', # (optional)
    # configuration items with changed time before then ....
    ConfigItemChangeTimeOlderDate => '2006-01-19 23:59:59', # (optional)

    What => [ # (optional)
        # each array element is a and condition
        {
            # or condition in hash
            "[%]{'ElementA'}[%]{'ElementB'}[%]{'Content'}" => '%contentA%',
            "[%]{'ElementA'}[%]{'ElementC'}[%]{'Content'}" => '%contentA%',
        },
        {
            "[%]{'ElementA'}[%]{'ElementB'}[%]{'Content'}" => '%contentB%',
            "[%]{'ElementA'}[%]{'ElementC'}[%]{'Content'}" => '%contentB%',
        },
        {
            # use array reference if different content with same key was searched
            "[%]{'ElementA'}[%]{'ElementB'}[%]{'Content'}" => ['%contentC%', '
↪%contentD%', '%contentE%'],
            "[%]{'ElementA'}[%]{'ElementC'}[%]{'Content'}" => ['%contentC%', '
↪%contentD%', '%contentE%'],
        },
    ],

    PreviousVersionSearch => 1, # (optional) default 0 (0|1)

    OrderBy => [ 'ConfigItemID', 'Number' ], # (optional)
    # default: [ 'ConfigItemID' ]
    # (ConfigItemID, Number, Name, ClassID, DeplStateID, InciStateID,
    # CreateTime, CreateBy, ChangeTime, ChangeBy)

    # Additional information for OrderBy:
    # The OrderByDirection can be specified for each OrderBy attribute.
    # The pairing is made by the array indices.

    OrderByDirection => [ 'Down', 'Up' ], # (optional)
    # default: [ 'Down' ]
    # (Down | Up)

    Limit      => 122, # (optional)
    UsingWildcards => 0, # (optional) default 1
);

```

ConfigItemUpdate () API

```

perform ConfigItemUpdate Operation. This will return the updated configuration item
↳number.

my $Result = $OperationObject->Run(
    Data => {
        UserLogin => 'some agent login',           # UserLogin or AccessToken is
        AccessToken => 123,                       # required

        Password => 'some password',             # if UserLogin is sent then
↳Password is required

        ReplaceExistingData => 0,                # optional, 0 or 1, default 0
↳existing XML data and attachments              # this will replace the

        ConfigItem => {
            Class => 'Configuration Item Class',
            Name => 'The Name',
            DeplState => 'deployment state',
            InciState => 'incident state',
            CIXMLData => $ArrayHashRef,          # it depends on the
↳Configuration Item class and definition

            Attachment => [
                {
                    Content => 'content'         # base64 encoded
                    ContentType => 'some content type'
                    Filename => 'some fine name'
                },
                # ...
            ],
            # or
            #Attachment => {
            # Content => 'content'
            # ContentType => 'some content type'
            # Filename => 'some fine name'
            #},
        },
    },
);

$Result = {
    Success => 1,                                # 0 or 1
    ErrorMessage => '',                          # in case of error
    Data => {                                     # result data payload after Operation
        ConfigItemID => 123,                      # Configuration Item ID number in
↳OTRS::ITSM (Service desk system)
        Number => 2324454323322                  # Configuration Item Number in
↳OTRS::ITSM (Service desk system)
        Error => {                                # should not return errors
            ErrorCode => 'ConfigItemUpdate.ErrorCode'
            ErrorMessage => 'Error Description'
        },
    },
},

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

};

2.1.2.2 Web-Services

Dieses Paket enthält einige neue Funktionen zum Erstellen, Ändern, Abrufen, Löschen und Suchen von Configuration Items über das Generic Interface. Die folgenden Operationen sind verfügbar:

- `ConfigItemCreate()`
- `ConfigItemDelete()`
- `ConfigItemGet()`
- `ConfigItemSearch()`
- `ConfigItemUpdate()`

Siehe auch:

Weitere Informationen finden Sie in der WSDL-Datei, die sich in `development/webservices/GenericConfigItemConnectorSOAP.wsdl` Ihrer Instanz befindet.

Neue Operationen

Diese neuen Operationen sind im Modul *Webservices* in der Gruppe *Prozess & Automation* verfügbar:

- `ConfigItem::ConfigItemCreate`
- `ConfigItem::ConfigItemDelete`
- `ConfigItem::ConfigItemGet`
- `ConfigItem::ConfigItemSearch`
- `ConfigItem::ConfigItemUpdate`

So nutzen Sie diese Operationen:

1. Einen Webservice hinzufügen oder bearbeiten.
2. Wählen Sie im Widget *OTRS als Provider* einen *Netzwerktransport* und speichern Sie den Webservice.
3. Die neuen Operationen sind im Feld *Operation hinzufügen* im Widget *OTRS als Provider* verfügbar.

Siehe auch:

Weitere Informationen finden Sie in den API-Referenzen im Kapitel *Prozessmanagement*.

Anwendungsbeispiele

Die folgenden Beispiele geben einen kurzen Einblick in die Verwendung der API für grundlegende Aktionen.

1. Configuration Item erstellen
 - URL: `/api/agent/config-item/create`
 - Methode: POST
 - Payload:

```
{
  "ConfigItem": {
    "Class": "Computer",
    "Name": "test name for new config item",
    "DeplState": "Production",
    "InciState": "Operational",
    "CIXMLData": {
      "Seriennummer": "SNR1"
      "NIC": {
        "NIC": "test",
        "IPoverDHCP": "Yes"
      }
    }
  }
}
```

2. Configuration Item aktualisieren

- URL: /api/agent/config-item/4/update, wobei 4 die ID des zu aktualisierenden Configuration Items ist
- Methode: POST
- Payload:

```
{
  "ConfigItemID": "4",
  "ConfigItem": {
    "Class": "Computer",
    "Name": "test name for new config item",
    "DeplState": "Production",
    "InciState": "Operational",
    "CIXMLData": {
      "Seriennummer": "SNR2"
      "NIC": {
        "NIC": "test",
        "IPoverDHCP": "Yes"
      }
    }
  }
}
```

Bemerkung: Die Klasse muss übertragen werden, hat aber keinen Einfluss auf das Configuration Item, wenn es aktualisiert wird. Wenn Sie einen Configuration Item in der Klasse `Location` aktualisieren und die Klasse `Computer` übertragen, bleibt der Configuration Item in der Klasse `Location`.

3. Configuration Item abrufen

- URL: /api/agent/config-item/4, wobei 4 die ID des abzurufenden Configuration Items ist
- Methode: GET

4. Configuration Items auflisten

- URL: /api/agent/config-item/list
- Methode: POST

2.1.3 Administration

Nach der Installation des Pakets stehen einige neue Klassen im *General Catalog* und ein neues Modul im Administrator-Interface zur Verfügung.

2.1.3.1 General Catalog

ITSM Configuration Management fügt einige neue Klassen zum *General Catalog* hinzu. Die Verwaltung des General Catalogs ist im Modul *General Catalog* in der Gruppe *Administration* verfügbar.

List
CATALOG CLASS
ITSM::ConfigItem::Class
ITSM::ConfigItem::Computer::Type
ITSM::ConfigItem::DeploymentState
ITSM::ConfigItem::Hardware::Type
ITSM::ConfigItem::Location::Type
ITSM::ConfigItem::Network::Type
ITSM::ConfigItem::Software::LicenceType
ITSM::ConfigItem::Software::Type
ITSM::ConfigItem::YesNo
ITSM::Core::IncidentState
ITSM::Service::Type
ITSM::SLA::Type

Abb. 7: Listenansicht des General Catalog

Neue Klassen

ITSM::ConfigItem::Class

Eine Klasse für Configuration Item-Klassen.

Siehe auch:

Die Klassendefinition von Configuration Item-Klassen kann im Modul *Configuration Items* in der Gruppe *CMDB Settings* verwaltet werden.

ITSM::ConfigItem::Computer::Type

Eine Klasse für Computer-Typen, die in *Configuration Items* beim Hinzufügen oder Bearbeiten von Configuration Items vom Typ „Computer“ ausgewählt werden kann.

ITSM::ConfigItem::DeploymentState

Eine Klasse für Bereitstellungs-Status, die in *Configuration Items* beim Hinzufügen oder Bearbeiten von Configuration Items ausgewählt werden kann.

ITSM::ConfigItem::Hardware::Type

Eine Klasse für Hardware-Typen, die in *Configuration Items* beim Hinzufügen oder Bearbeiten von Configuration Items vom Typ „Hardware“ ausgewählt werden kann.

ITSM::ConfigItem::Location::Type

Eine Klasse für Standort-Typen, die in *Configuration Items* beim Hinzufügen oder Bearbeiten von Configuration Items vom Typ „Standort“ ausgewählt werden kann.

ITSM::ConfigItem::Network::Type

Eine Klasse für Netzwerktypen, die in *Configuration Items* beim Hinzufügen oder Bearbeiten von Configuration Items vom Typ „Netzwerk“ ausgewählt werden kann.

ITSM::ConfigItem::Software::LicenceType

Eine Klasse für Software-Lizenztypen, die in *Configuration Items* beim Hinzufügen oder Bearbeiten von Configuration Items vom Typ „Software“ ausgewählt werden kann.

ITSM::ConfigItem::Software::Type

Eine Klasse für Software-Typen, die in *Configuration Items* beim Hinzufügen oder Bearbeiten von Configuration Items vom Typ „Software“ ausgewählt werden kann.

ITSM::ConfigItem::YesNo

Diese Klasse beinhaltet die Einträge *Ja* und *Nein*.

2.1.3.2 Import und Export

Verwenden Sie diese Ansicht, um Import- und Exportvorlagen zu erstellen. Die Ansicht zur Verwaltung der Import/Exportvorlagen ist im Modul *Import/Export* in der Gruppe *Administration* verfügbar.

Import/Export Management

Actions		Config Item						
<input type="button" value="Add template"/>		NUMBER	NAME	FORMAT	VALIDITY	DELETE	START IMPORT	START EXPORT
Note		000002	Test	CSV	valid		Import	Export

Abb. 8: Import/Exportvorlagen verwalten

Import/Exportvorlagen verwalten

So erstellen Sie eine neue Vorlage:

1. Klicken Sie in der linken Seitenleiste auf die Schaltfläche *Vorlage hinzufügen*.
2. Füllen Sie die benötigten Felder in allen Schritten aus.
3. Klicken Sie auf die Schaltfläche *Abschließen*.

Step 1 of 5 - Edit common information:

Name:

Object:

Format:

Valid:

Comment:

or [Cancel](#)

Abb. 9: Neue Import/Exportvorlage erstellen

So bearbeiten Sie eine Vorlage:

1. Klicken Sie in der Liste mit den Vorlagen auf eine Vorlage.
2. Ändern Sie die Felder in allen Schritten.
3. Klicken Sie auf die Schaltfläche *Abschließen*.

Step 1 of 5 - Edit common information:

Name:

Object: ITSMConfigItem

Format: CSV

Valid:

Comment:

or

Abb. 10: Import/Exportvorlage bearbeiten

So löschen Sie eine Vorlage:

1. Klicken Sie in der Liste mit den Vorlagen auf das Papierkorb-Symbol.
2. Klicken Sie auf die Schaltfläche *Bestätigen*.

Config Item						
NUMBER	NAME	FORMAT	VALIDITY	DELETE	START IMPORT	START EXPORT
000002	Test	CSV	valid		Import	Export

Abb. 11: Import/Exportvorlage löschen

So importieren Sie Daten basierend auf einer Vorlage:

1. Klicken Sie in der Liste mit den Vorlagen auf die Verknüpfung *Importieren*.
2. Klicken Sie auf die Schaltfläche *Durchsuchen...* und wählen Sie eine CSV-Datei.
3. Klicken Sie auf die Schaltfläche *Import starten*.

Import information:

Name: Test

Source File: Nincs kijelölve fájl.

Abb. 12: Daten importieren

So exportieren Sie Daten basierend auf einer Vorlage:

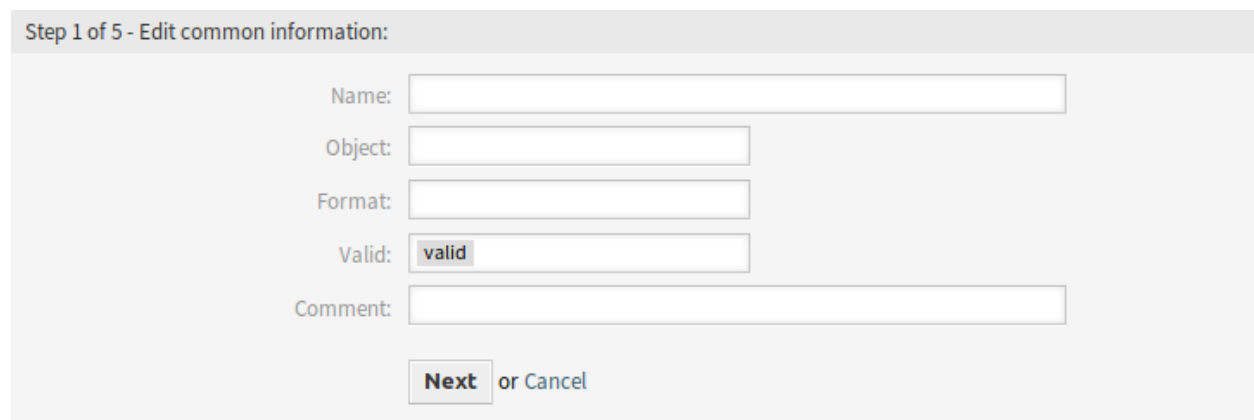
1. Klicken Sie in der Liste mit den Vorlagen auf die Verknüpfung *Exportieren*.
2. Wählen Sie einen Speicherort auf Ihrem Computer um die `Export_ACL-ym` Datei zu speichern.

Einstellungen für Import/Exportvorlagen

Die folgenden Einstellungen sind verfügbar, wenn Sie diese Ressource hinzufügen oder bearbeiten. Die mit einem Sternchen gekennzeichneten Felder sind Pflichtfelder.

Bemerkung: Das Import/Export-Paket ist unabhängig. Das bedeutet, dass die folgenden Einstellungen unterschiedlich sein können, wenn keine Konfigurationselemente importiert oder exportiert werden.

Gemeinsame Informationen bearbeiten



Step 1 of 5 - Edit common information:

Name:

Object:

Format:

Valid:

Comment:

or

Abb. 13: Gemeinsame Informationen bearbeiten

Name *

Der Name der Ressource. In dieses Feld können beliebige Zeichen eingegeben werden, einschließlich Großbuchstaben und Leerzeichen. Der Name wird in der Übersichtstabelle angezeigt.

Objekt *

Wählen Sie den Objekttyp aus, den Sie importieren oder exportieren möchten.

Format *

Wählen Sie das Import und Exportformat.

Gültigkeit *

Setzt die Gültigkeit dieser Ressource. Jede Ressource kann nur in OTRS verwendet werden, wenn dieses Feld auf *gültig* gesetzt ist. Wenn Sie dieses Feld auf *ungültig* oder *ungültig-temporär* setzen, wird die Nutzung der Ressource deaktiviert.

Kommentar

Hier können Sie der Ressource zusätzliche Informationen hinzufügen. Es wird empfohlen, dieses Feld als Beschreibung der Ressource zur besseren Übersichtlichkeit immer mit einem vollständigen Satz zu füllen, da der Kommentar auch in der Übersichtstabelle angezeigt wird.

Objektinformationen bearbeiten

Step 2 of 5 - Edit object information:

Name: Test

Object: ITSMConfigItem

Class:

Maximum number of one element:

Empty fields indicate that the current values are kept:

Abb. 14: Objektinformationen bearbeiten

Name

Das ist ein „Nur lesen“-Feld aus dem vorherigen Schritt. Verwenden Sie die Schaltfläche *Zurück*, um es zu bearbeiten.

Objekt

Das ist ein „Nur lesen“-Feld aus dem vorherigen Schritt. Verwenden Sie die Schaltfläche *Zurück*, um es zu bearbeiten.

Klasse *

Wählen Sie die Klasse aus, die vom Import und Export betroffen sein soll.

Maximale Anzahl eines Elements *

Geben Sie die maximale Anzahl von Spalten pro Array-Attribut an, die beim Mapping des Imports aus oder Exports in die CSV-Datei abgebildet werden können.

Leere Felder zeigen an, dass die aktuellen Werte beibehalten werden

Wählen Sie dieses Kontrollkästchen, wenn das leere Feld die Daten in OTRS behalten soll. Andernfalls werden die Daten mit einem leeren Wert überschrieben.

Format-Informationen bearbeiten

Name

Das ist ein „Nur lesen“-Feld aus dem vorherigen Schritt. Verwenden Sie die Schaltfläche *Zurück*, um es zu bearbeiten.

Format

Das ist ein „Nur lesen“-Feld aus dem vorherigen Schritt. Verwenden Sie die Schaltfläche *Zurück*, um es zu bearbeiten.

Spaltentrenner *

Wählen Sie einen Spaltentrenner für die CSV-Datei.

Zeichensatz

Wählen Sie eine Zeichenkodierung für die CSV-Datei aus.

Mit Spaltenüberschriften

Legen Sie fest, ob Spaltenüberschriften dargestellt werden sollen oder nicht.

Step 3 of 5 - Edit format information:

Name: Test

Format: CSV

Column Separator:

Charset: UTF-8

Include Column Headers: No

Abb. 15: Format-Informationen bearbeiten

Mapping-Informationen bearbeiten

Step 4 of 5 - Edit mapping information:

Name: Test Object: Config Item Format: CSV

KEY	IDENTIFIER	COLUMN	UP	DOWN	DELETE
No map elements found.					

Abb. 16: Mapping-Informationen bearbeiten

Klicken Sie auf die Schaltfläche *Mapping-Element hinzufügen*, um ein Element aus der Klasse hinzuzufügen. Sie können auch angeben, ob dieses Element ein Identifikator ist. Die Reihenfolge der Elemente ist sortierbar.

Suchinformationen bearbeiten

Name der Vorlage

Das ist ein „Nur lesen“-Feld aus dem vorherigen Schritt. Verwenden Sie die Schaltfläche *Zurück*, um es zu bearbeiten.

Export per Suche einschränken

Sie können für jedes Attribut der ausgewählten Klasse einen Suchbegriff hinzufügen, um die Import- und Exportfunktionen einzuschränken. Unterhalb dieses Feldes sind die möglichen Felder aufgelistet.

Bemerkung: Die anderen Felder stammen aus dem Backend-Treiber und können je nach dem verwendeten zu importierenden oder zu exportierenden Objekt unterschiedlich sein.

Step 5 of 5 - Edit search information:

Template Name: **Test**

Restrict export per search:

Number:

Name:

Deployment State:

Incident State:

Vendor:

Model:

Description:

Type:

Serial Number:

FQDN:

Network Adapter::IP Address:

Note:

Abb. 17: Suchinformationen bearbeiten

2.1.3.3 Systemkonfiguration

Anzeige der klassenspezifischen Spalten des Configuration Items

Klassenspezifische Spalten des Configuration Items (z. B. die Kapazität einer Festplatte) werden in der Konfigurationsübersichtsliste und in der Organizer-Elementliste des Configuration Items standardmäßig nicht angezeigt.

Um die Feldwerte des Configuration Items als Tabellenspalten darzustellen, muss die YAML-Konfiguration der Listen erweitert werden.

Das folgende Beispiel zeigt, wie Sie das Feld `Computer::HardDisk::1` und `Computer::HardDisk::1::Capacity::1` der Klasse `Computer` zur Übersichtsliste der Configuration Items hinzufügen:

1. Suchen Sie nach der Einstellung `AgentFrontend::ConfigItemList###DefaultConfig`.
2. Ergänzen Sie die YAML-Konfiguration wie folgt:

```
Columns:
  Computer::HardDisk::1:
    isVisible: 2
  Computer::HardDisk::1::Capacity::1:
    isVisible: 2
```



Abb. 18: Beispiel YAML-Konfiguration der Configuration Item-Übersicht

3. Nehmen Sie die geänderte Konfiguration in Betrieb.

The screenshot shows the 'Configuration Items (3 Configuration Items)' overview table. The table has the following columns: Class, Name, Incident State, Deployment State, ConfigItem#, Computer: Hard Disk 1, Computer: Capacity, Changed, and Duplicate. The data rows are:

Class	Name	Incident State	Deployment State	ConfigItem#	Computer: Hard Disk 1	Computer: Capacity	Changed	Duplicate
Hardware	Keyboard Logitech	Operational	Production	5465000001			2 hours ago	
Computer	Poseidon	Operational	Production	5464000002	Samsung 460 Evo 2.5"	500 GB	2 hours ago	
Computer	Zeus	Operational	Production	5464000001	Samsung 860 Evo 2.5"	1000 GB	23 hours ago	

Abb. 19: Configuration Item Übersicht mit klassenspezifischen Spalten

Anwenden von klassenspezifischen Filtern für Configuration Items

Das folgende Beispiel zeigt, wie Sie den Filter für das Feld `Computer::Model` der Klasse `Computer` in der Übersichtsliste der Configuration Items anwenden.

Zunächst müssen Sie sicherstellen, dass der entsprechende Klassenfilter auch angewendet wird, sonst werden alle klassenspezifischen Filter einfach ignoriert.

Dies kann über den Filter `ClassIDs` erfolgen, der die ID der Klasse als Wert annimmt. Bitte führen Sie die folgenden Schritte aus, um die ID der Klasse zu erhalten und die Filter anzuwenden.

1. Rufen Sie im Administrator-Interface die Ansicht *Configuration Items* - Verwaltung auf.
2. Klicken Sie in der Liste auf die entsprechende Klasse.

Die `ClassID` wird nun in der URL angezeigt, in diesem Fall ist die ID zum Beispiel 22:

```
``otrs/index.pl/?Action=AdminITSMConfigItem;Subaction=DefinitionList;ClassID=22``
```

3. Suchen Sie nach der Einstellung `AgentFrontend::ConfigItemList###DefaultConfig`.
4. Setzen Sie sowohl den Filter für die Klasse auf den in Schritt 2 ermittelten Wert als auch den Feldfilter auf den gewünschten Wert:

```
ActiveFilters:
  ClassIDs:
    Value:
      - 22
  Computer::Model:
    Value: ModelA
```

5. Nehmen Sie die geänderte Konfiguration in Betrieb.

Bemerkung: Die zu filternden Felder müssen auf `Searchable: 1` in ihren Klassendefinitionen gesetzt sein. Siehe [here](#) für weitere Informationen.

Anwenden von Configuration Item-Filtern für alle Klassen

Das folgende Beispiel zeigt, wie Sie die Filter für die gemeinsamen Felder `Owner` und `CustomerID` anwenden, die in allen Klassen der Configuration Item-Übersichtsliste verwendet werden.

1. Suchen Sie nach der Einstellung `AgentFrontend::ConfigItemList###DefaultConfig`.
2. Ergänzen Sie die YAML-Konfiguration wie folgt:

```
AvailableSearchInAllClassesFilters:
- Owner
- CustomerID
```

3. Nehmen Sie die geänderte Konfiguration in Betrieb.

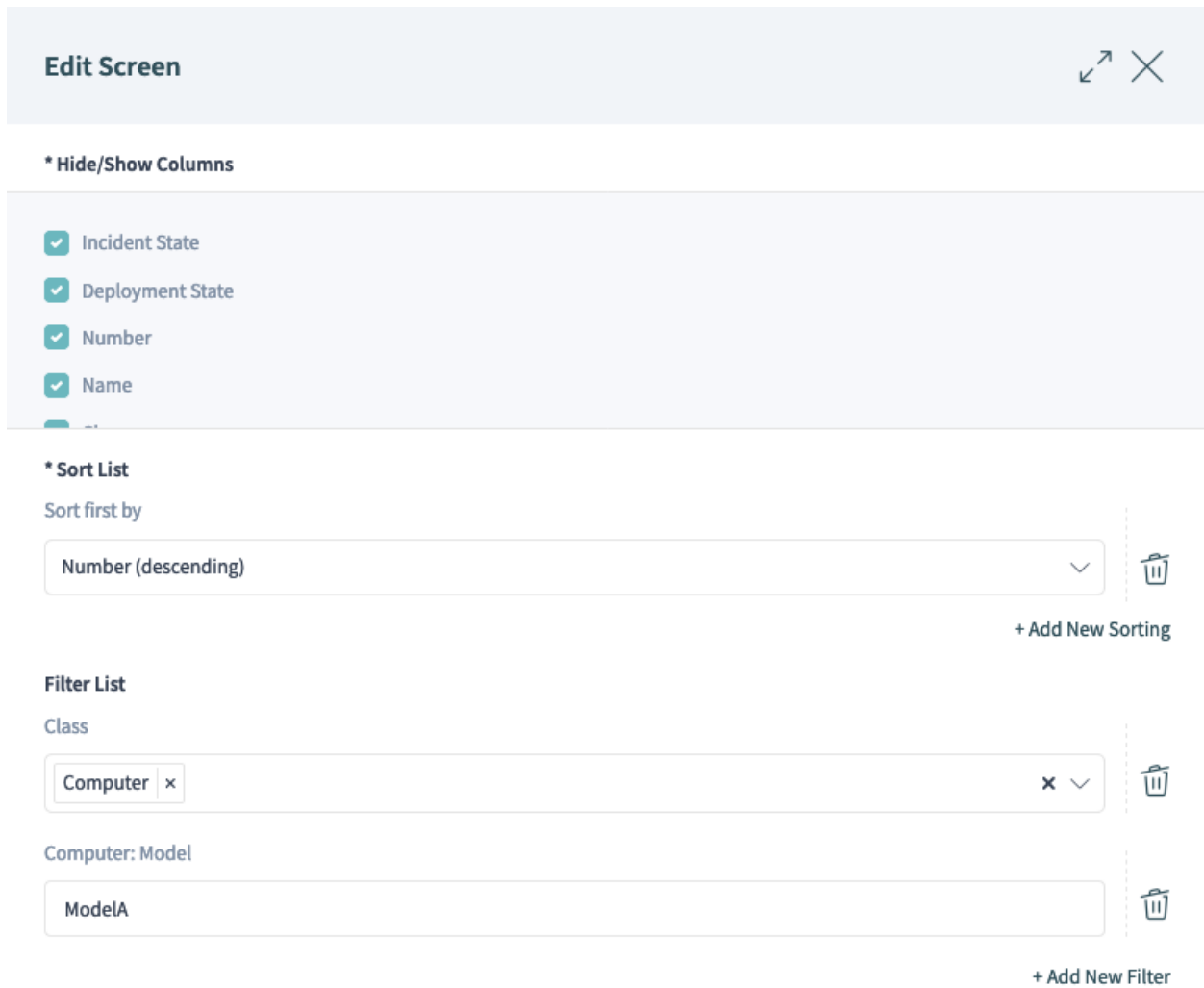


Abb. 20: Configuration Item Übersicht mit klassenspezifischen Filtern

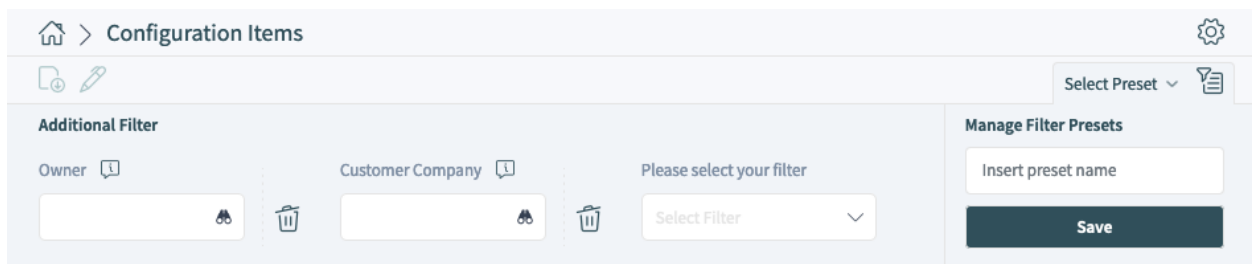


Abb. 21: Configuration Item-Übersicht mit allen Klassenfiltern

2.1.4 CMDB-Einstellungen

Nach der Installation des Pakets ist eine neue Gruppe *CMDB Einstellungen* mit einem neuen Modul im Administrator-Interface verfügbar.

2.1.4.1 Configuration Items

Verwenden Sie diese Ansicht, um die Klassendefinition von Configuration Items zu verwalten. Die Ansicht zur Verwaltung der Configuration Item-Klassen ist im Modul *Config Items* in der Gruppe *CMDB-Einstellungen* verfügbar.

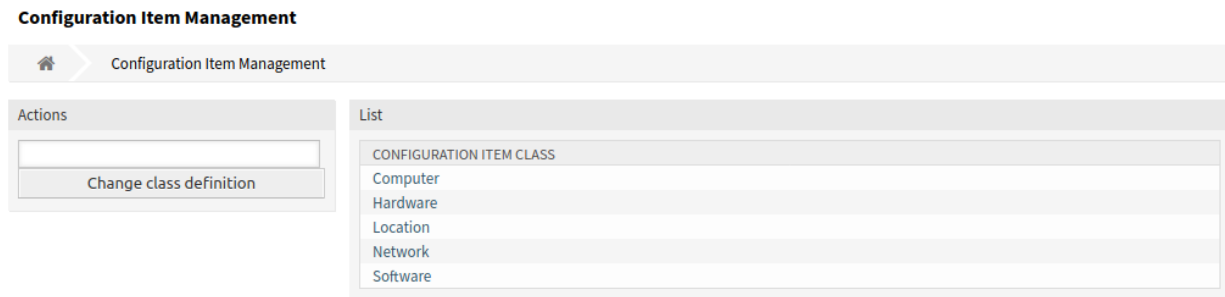


Abb. 22: Configuration Item Management

So können Sie eine Configuration Item-Klassendefinition hinzufügen oder bearbeiten:

1. Wählen Sie eine Klasse aus dem Dropdown-Menü in der linken Seitenleiste.
2. Klicken Sie auf die Schaltfläche *Klassendefinition ändern*.
3. Fügen Sie die Klassendefinition im YAML-Format hinzu oder bearbeiten Sie sie.
4. Klicken Sie auf die Schaltfläche *Speichern* oder *Speichern und abschließen*.

So schauen Sie sich die Klassendefinition eines Configuration Items an:

1. Klicken Sie in der Liste mit den Klassen auf eine Klasse.
2. Wählen Sie eine Version, indem Sie auf einen Klassennamen in der Liste der Klassenversionen klicken.

Siehe auch:

Neue Configuration Item-Klassen können im Modul [General Catalog](#) in der Administratoroberfläche hinzugefügt werden.

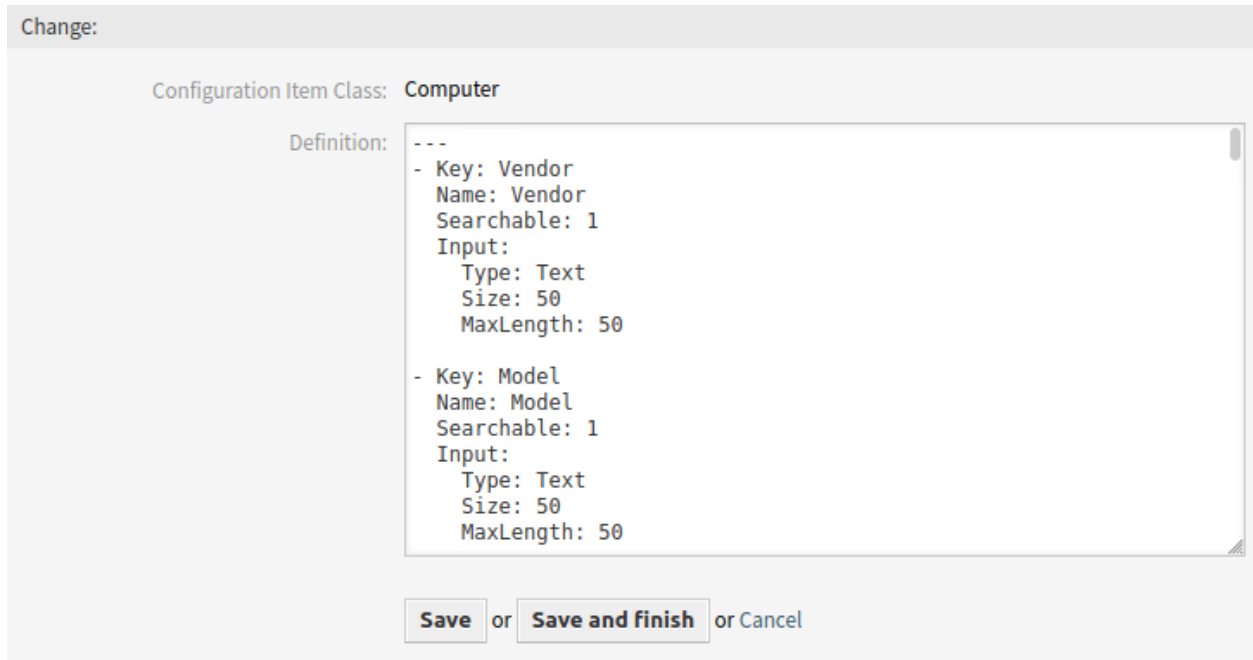


Abb. 23: Configuration Item - Klassendefinition bearbeiten

Configuration Item Management

Configuration Item Management > Computer

Actions

Computer x

Change class definition

Go to overview

List

CONFIGURATION ITEM CLASS	VERSION	CREATED BY	CREATED
Computer	1	Admin OTRS	09/15/2020 10:10:57 (Europe/Budapest)

Abb. 24: Configuration Item - Klassenversionen

Klassendefinitions-Typen und Formularelemente

Bei der Definition einer Klasse können mehrere Eingabefeldtypen verwendet werden. Diese Eingabefeldtypen werden verwendet, um das Editierformular für das Anlegen neuer oder das Bearbeiten bereits vorhandener Configuration Items zu generieren.

Der folgende Block ist ein Beispiel für ein Formularefeld namens *Operating System*.

```

---
- Key: OperatingSystem
  Name: Operating System
  Input:
    Type: Text
    Size: 50
    MaxLength: 100

```

Die folgenden Einstellungen sind verfügbar, wenn Sie diese Ressource hinzufügen oder bearbeiten. Die mit einem Sternchen gekennzeichneten Felder sind Pflichtfelder.

Key *

Muss eindeutig sein und darf nur alphabetische und numerische Zeichen enthalten. Wenn dies geändert wird, sind Daten aus alten Definitionen nicht mehr lesbar.

Name *

Die Bezeichnung des Feldes im Formular. In dieses Feld können beliebige Zeichen eingegeben werden, einschließlich Großbuchstaben und Leerzeichen.

Bemerkung: Es wird empfohlen, für Namen immer englische Wörter zu verwenden.

Siehe auch:

Namen können mit benutzerdefinierten Sprachdateien in andere Sprachen übersetzt werden. Weitere Informationen finden Sie im Kapitel [Benutzerdefinierte Sprachdatei](#) im Administrationshandbuch.

Searchable

Legt fest, ob das Feld durchsuchbar ist oder nicht. Mögliche Werte sind *0* oder *1*.

Input *

Leitet die Definition des Eingabefeldes ein. Ein Eingabefeld kann die folgenden Attribute enthalten:

Type *

Definiert den Typ des Elements. Muss als logischer Block eingerückt platziert werden. Mögliche Werte sind:

- **Customer:** Eine Dropdown-Liste zur Auswahl eines Kundenbenutzers aus dem Datenbank-Backend. Das Feld kann mit Wildcards (*) verwendet werden.
- **CustomerCompany:** Eine Dropdown-Liste zur Auswahl eines Kunden aus dem Datenbank-Backend.
- **Date:** Ein Feld zur Auswahl eines Datums.
- **DateTime:** Ein Feld zur Auswahl von Datum und Uhrzeit.
- **Dummy:** Dieses Feld wird verwendet, um den anderen Elementen eine Struktur zu geben. Es hat normalerweise Sub-Strukturen.
- **GeneralCatalog:** Eine Dropdown-Liste zur Auswahl einer allgemeinen General Catalog-Klasse. Die General Catalog-Klasse muss definiert werden, bevor sie als Eingabeart

verwendet wird. Die Elemente der General Catalog-Klasse sind die Optionen der Dropdown-Liste.

- **Integer**: Eine Dropdown-Liste mit Ganzzahlen.
- **Text**: Ein einzeliges Textfeld.
- **TextArea**: Ein mehrzeiliges Textfeld.

Required

Legt fest, ob das Feld ein Pflichtfeld sein soll oder nicht. Mögliche Werte sind *0* oder *1*.

Size

Legt die Größe des Textfeldes fest. Der Wert muss eine positive ganze Zahl sein.

MaxLength

Definiert die maximale Anzahl von Zeichen, die in das Textfeld eingegeben werden können. Der Wert muss eine positive ganze Zahl sein.

RegEx

Ein regulärer Ausdruck, um die möglichen Werte des Textfeldes einzuschränken.

RegExErrorMessage

Die angezeigte Fehlermeldung, wenn die Eingabe nicht mit der im regulären Ausdruck angegebenen Definition übereinstimmt.

Class

Der Name der Klasse, die für die Dropdown-Liste verwendet werden soll. Erforderlich für den Typ `GeneralCatalog`.

Translation

Legt fest, ob die Einträge des General Catalogs übersetzt werden müssen. Mögliche Werte sind: *0* oder *1*.

YearPeriodPast

Legt fest, wie viele Jahre in der Vergangenheit ausgehend vom aktuellen Jahr in einem Datums- oder Datums-/Uhrzeitfeld zur Auswahl stehen. Der Wert muss eine positive ganze Zahl sein.

YearPeriodFuture

Legt fest, wie viele Jahre in der Zukunft ab dem aktuellen Jahr in einem Datums- oder Datum/Uhrzeit-Feld zur Auswahl stehen. Der Wert muss eine positive Ganzzahl sein.

ValueMin

Definiert den Minimalwert für ein Integerfeld.

ValueMax

Definiert den Maximalwert für ein Integerfeld.

ValueDefault

Definiert den Maximalwert für ein Integerfeld.

CountMin

Definiert, wie viele Felder mindestens für einen Eingabefeld-Typ verfügbar sind. Der Wert muss eine positive ganze Zahl sein.

CountMax

Definiert, wie viele Felder mindestens für einen Eingabefeld-Typ verfügbar sind. Der Wert muss eine positive ganze Zahl sein.

CountDefault

Legt fest, wie viele Felder standardmäßig angezeigt werden sollen. Der Wert muss eine positive ganze Zahl sein.

Sub

Definiert ein Unterelement im Eingabefeld. Das Unterelement kann wieder eigene Eingabefelder enthalten. Es ist nützlich, wenn Sie bestimmte Eigenschaften unter einer Haupteigenschaft haben.

SuppressVersionAdd

Dadurch kann das Anlegen einer neuen Version eines Configuration Items unterdrückt werden, wenn sich ein Attribut geändert hat. Mögliche Werte sind `UpdateLastVersion` und `Ignore`.

- `UpdateLastVersion`: Wenn dieser Wert gesetzt ist und es kein anderes aktualisiertes Attribut gibt, wird das Attribut in der aktuellen Version aktualisiert, ohne eine neue Version anzulegen.
- `Ignore`: Wenn dieser Wert gesetzt ist und es kein anderes aktualisiertes Attribut gibt, wird nichts getan und es wird keine neue Version erstellt.

Klassendefinition-Referenz

Die folgende Klassendefinition ist ein Beispiel für alle möglichen Optionen.

Bemerkung: `CustomerID` und `Owner` sind Sonderschlüssel, da diese Schlüssel in *Kunden* und *Kundenbenutzer* verwendet werden, um Configuration Items standardmäßig automatisch Kunden und Kundenbenutzern zuzuordnen.

```

---
- Key: OperatingSystem
  Name: Operating System
  Searchable: 1
  Input:
    Type: Text
    Required: 1
    Size: 50
    MaxLength: 100
    RegEx: Linux|MacOS|Windows|Other
    RegExErrorMessage: The operating system is unknown.
  CountMin: 0
  CountMax: 5
  CountDefault: 1

- Key: Description
  Name: Description
  Searchable: 0
  Input:
    Type: TextArea
    Required: 0
  CountMin: 0
  CountMax: 1
  CountDefault: 0

- Key: Type
  Name: Type
  Searchable: 1
  Input:
    Type: GeneralCatalog
    Class: ITSM::ConfigItem::Software::Type
    Required: 1
    Translation: 1

```

(Fortsetzung auf der nächsten Seite)

```
- Key: CustomerID
Name: Customer Company
Searchable: 1
Input:
  Type: CustomerCompany

- Key: Owner
Name: Owner
Searchable: 1
Input:
  Type: Customer

- Key: LicenseKey
Name: License Key
Searchable: 1
Input:
  Type: Text
  Size: 50
  MaxLength: 50
  Required: 1
CountMin: 0
CountMax: 100
CountDefault: 0
Sub:
- Key: Quantity
Name: Quantity
Input:
  Type: Integer
  ValueMin: 1
  ValueMax: 1000
  ValueDefault: 1
  Required: 1
CountMin: 0
CountMax: 1
CountDefault: 0

- Key: ExpirationDate
Name: Expiration Date
Input:
  Type: Date
  Required: 1
  YearPeriodPast: 20
  YearPeriodFuture: 10
CountMin: 0
CountMax: 1
CountDefault: 0

- Key: LastUsed
Name: Last Used
Input:
  Type: DateTime
  Required: 1
CountMin: 0
CountMax: 1
CountDefault: 0
SuppressVersionAdd: UpdateLastVersion
```

2.2 Agenten-Interface

Dieses Kapitel beschreibt die neuen Funktionen, die im Agenten-Interface nach der Installation des Paketes verfügbar sind.

2.2.1 Configuration Items

Nach der Installation des Pakets ist im Hauptmenü ein neuer Menüabschnitt verfügbar.

Bemerkung: Um Benutzern Zugriff auf das Menü *Asset-Verwaltung* zu gewähren, müssen sie als Mitglied zur Gruppe *itsm-configitem* hinzugefügt werden.

2.2.1.1 Configuration Item erstellen

Verwenden Sie diese Ansicht, um neue Configuration Items zur Configuration Management Database hinzuzufügen.

So fügen Sie ein Configuration Item hinzu:

1. Wählen Sie eine Klasse aus der Liste mit den Klassen.
2. Füllen Sie die Pflichtfelder aus.
3. Klicken Sie auf die Schaltfläche *Erstellen*.

General Information

* Class
Select... ▼

* Name
[Text Input Field]

* Incident State ⓘ
Select... ▼

* Deployment State ⓘ
Select... ▼

Create

Abb. 25: Configuration Item erstellen

Siehe auch:

Die Felder in der Ansicht *Eigenschaften* können je nach Klasse unterschiedlich sein. Um die verfügbaren Felder zu sehen, schauen Sie bitte im Modul *Configuration Items* im Administrator-Interface.

Warnung: Die maximale Anzahl von 20000 Configuration Items sollte nicht überschritten werden. Ein Überschreiten dieser Grenze kann die Systemleistung beeinträchtigen.

2.2.1.2 Configuration Item-Liste

Diese Ansicht gibt einen Überblick über die Configuration Items. Configuration Items haben eine Spalte *Vorfallstatus*, die zwei Statusarten enthält:

- Operativ
- Vorfall

Für jeden Statustyp kann eine beliebige Anzahl von Status registriert werden. Der Status eines Configuration Items wirkt sich auf den Service-Status aus, der dynamisch berechnet und in der Ansicht *Services* im Agenten-Interface angezeigt wird.

Siehe auch:

Um die dynamische Berechnung zu aktivieren, aktivieren Sie die folgenden Einstellungen der Systemkonfiguration:

- ITSMConfigItem::SetIncidentStateOnLink
- ITSMConfigItem::LinkStatus::TicketTypes

<input type="checkbox"/> Incident State	Deployment State	ConfigItem#	Name	Class	Changed	Duplicate
<input type="checkbox"/> Operational	Production	1022000001	Web server	Computer	2 hours ago	

Abb. 26: Configuration Item-Liste

2.2.1.3 Configuration Item-Detailansicht

Verwenden Sie diese Ansicht, um die Details eines Configuration Items zu sehen. Die Configuration Item-Detailansicht ist verfügbar, wenn Sie ein Configuration Item aus einer Configuration Item-Liste auswählen.

Widgets in der Configuration Item-Detailansicht

Wie andere Business-Objekt-Detailansichten ist auch die Configuration Item-Detailansicht in hohem Maße anpassbar. Einige der folgenden Widgets werden bei der Standardinstallation angezeigt, andere müssen jedoch in der Ansichtskonfiguration hinzugefügt werden.

Widget „Configuration Item Information“

Dieses Widget zeigt Informationen über das Configuration Item an.

Widget „Configuration Item-Details“

Dieses Widget zeigt die Versionen der Configuration Items an. Jede Änderung an einem Configuration Item erzeugt eine neue Version. Wenn Sie auf eine Version in diesem Widget klicken, werden die Details angezeigt.

Configuration Item Information

ConfigItem# 1022000001

Name	Class	Incident State
Web server	Computer	Operational
Deployment State	Created	Changed
Production	2 hours ago	2 hours ago

Abb. 27: Widget „Configuration Item Information“

Configuration Item Version Details

Expand All | Select Preset

Deployment State	Incident State	Version#	Name	Created	Created By
Production	Operational	1	Web server	2 hours ago	Admin OTRS

Abb. 28: Widget „Configuration Item-Details“

Widget „Anhänge“

Dieses Widget kann verwendet werden, um Anhänge für Configuration Items anzuzeigen. Die Anhänge können heruntergeladen werden und für die Bilder wird eine Vorschaufunktion unterstützt.

Attachments

Select Preset

Type	Filename	File size	Preview	Download
	ryzen-3700x.pdf	472.33 KB		

Abb. 29: Widget „Anhänge“

Aktionen in der Configuration Item-Detailansicht

Die folgenden Aktionen sind in der Ticket-Detailansicht verfügbar.

Organisation

Diese Spalte fasst die folgenden Aktionen zusammen:

Configuration Item bearbeiten

Diese Aktion ermöglicht es Agenten, das Configuration Item zu bearbeiten.

Objekte verknüpfen

Diese Aktion ermöglicht es Agenten, andere Business-Objekte mit dem Configuration Item zu verknüpfen.

Configuration Item duplizieren

Diese Aktion ermöglicht es Agenten, das Configuration Item zu duplizieren.

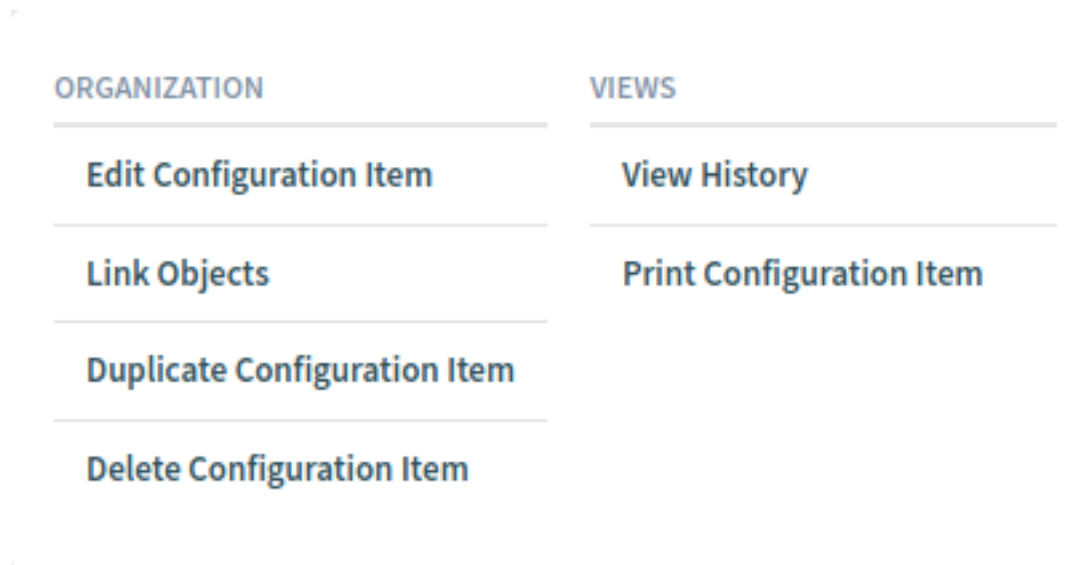


Abb. 30: Aktionen in der Configuration Item-Detailansicht

Configuration Item löschen

Diese Aktion ermöglicht es Agenten, das Configuration Item zu löschen.

Ansichten

Diese Spalte fasst die folgenden Aktionen zusammen:

Historie anzeigen

Diese Aktion ermöglicht es den Agenten, die Historie des Configuration Item einzusehen. Die Historie enthält alle Vorgänge, die in der Vergangenheit mit dem Configuration Item durchgeführt wurden, zusammen mit Zeitstempel und Benutzername der Person, die die Aktion durchgeführt hat.

Configuration Item ausdrucken

Diese Aktion ermöglicht es den Agenten, die Daten von Configuration Item als PDF-Datei auszudrucken und herunterzuladen.

2.2.2 Kunden

Nach der Installation des Pakets steht ein neues Widget mit dem Namen *Configuration Item-Liste* in der Kunden-Detailansicht zur Verfügung.

Configuration Item List (1 Configuration Item)						Select Preset ▾		
Incident State	Deployment State	ConfigItem# ▾	Name	Class	Changed			
Operational	Production	1022000001	Web server	Computer	20 minutes ago			

Abb. 31: Widget „Configuration Item-Liste“

Dieses Widget zeigt die Configuration Items, die dem Kunden zugeordnet sind.

Die Zuordnung erfolgt standardmäßig mit dem Attribut `CustomerID`. Wenn das Configuration Item ein anderes Attribut für die Verknüpfung verwendet, sollten Sie es in den Systemeinstellungen ändern.

Siehe auch:

Schauen Sie in der Systemkonfigurations-Einstellung `AgentFrontend::CustomerCompanyDetailView::WidgetType` für weitere Informationen.

Die Standardeinstellung ist:

```
ClassBasedCustomerIDSearch:
  Computer: CustomerID
  Hardware: CustomerID
  Location: CustomerID
  Network: CustomerID
  Software: CustomerID
ClassBasedCustomerSearch:
  Computer: CustomerID
  Hardware: CustomerID
  Location: CustomerID
  Network: CustomerID
  Software: CustomerID
```

Sie müssen auch das Attribut `Owner` in der Klassendefinition haben, um die zugeordneten Configuration Items anzuzeigen. Überprüfen Sie die vorhandenen Klassendefinitionen im Modul *Configuration Items*.

Wenn Ihre Klassendefinition nicht das Attribut `CustomerID` enthält, dann müssen Sie es manuell hinzufügen.

```
- Key: CustomerID
  Name: Customer Company
  Searchable: 1
  Input:
    Type: CustomerCompany
```

2.2.3 Kundenbenutzer

Nach der Installation des Pakets steht ein neues Widget mit dem Namen *Configuration Item-Liste* in der Kundenbenutzer-Detailansicht zur Verfügung.

Incident State	Deployment State	Configitem#	Name	Class	Changed
Operational	Production	1022000001	Web server	Computer	20 minutes ago

Abb. 32: Widget „Configuration Item-Liste“

Dieses Widget zeigt die Configuration Items, die dem Kunden zugeordnet sind.

Die Zuordnung erfolgt standardmäßig mit dem Attribut `Owner`. Wenn das Configuration Item ein anderes Attribut für die Verknüpfung verwendet, sollten Sie es in den Systemeinstellungen ändern.

Siehe auch:

Siehe Systemkonfigurations-Einstellung `AgentFrontend::CustomerUserDetailView::WidgetType###ConfigItem` für weitere Informationen.

Die Standardeinstellung ist:

ClassBasedCustomerUserSearch:

Computer: Owner
Hardware: Owner
Location: Owner
Network: Owner
Software: Owner

ClassBasedCustomerSearch:

Computer: Owner
Hardware: Owner
Location: Owner
Network: Owner
Software: Owner

Sie müssen auch dieses Attribut `Owner` in der Klassendefinition haben, um die zugeordneten Configuration Items anzuzeigen. Überprüfen Sie die vorhandenen Klassendefinitionen im Modul [Configuration Items](#).

Wenn Ihre Klassendefinition nicht das Attribut `Owner` enthält, dann müssen Sie es manuell hinzufügen.

```
- Key: Owner  
Name: Owner  
Searchable: 1  
Input:  
  Type: Customer
```

2.3 Externes Interface

Dieses Paket hat kein externes Interface.