



Vancouver Now Platform Capabilities

Vancouver Now Platform Capabilities

Last updated: October 25, 2023

Some examples and graphics depicted herein are provided for illustration only. No real association or connection to ServiceNow products or services is intended or should be inferred.

This PDF was created from content on docs.servicenow.com. The web site is updated frequently. For the most current ServiceNow product documentation, go to docs.servicenow.com.

Company Headquarters

2225 Lawson Lane
Santa Clara, CA 95054
United States
(408)501-8550

Configuration Management Database (CMDB)

The CMDB is a centralized source that gives you full visibility into your IT environment. By storing information about your organization's infrastructure and how it is configured, this system allows you to monitor your network and ensure stability and best performance.

Overview

[Video: CMDB | Overview](#)

Managing the CMDB



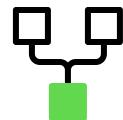
Build, adjust, and monitor representations of your business infrastructure to support ServiceNow® products and services.

Common Service Data Model (CSDM)



Get consistent modeling in the CMDB, with standardized data terminology that can be used across the entire Now Platform.

Dependency Views



View graphical data of your infrastructure's elements and how they connect to and support the rest of your network.

Application Services



Create application services Cls from devices and applications that business units and different products can use.

Data Certification



Verify all your data in the CMDB and keep regular checkups using this powerful application.

Managing the CMDB

Create models of your infrastructure using Configuration Management. Store your infrastructure data, represented by configuration items (CI)—all the computers, servers, routers, database instances, and services in your network. Analyze trends, reduce problems, and handle incidents by monitoring relationships between CI.

Common Service Data Model (CSDM)

Ensure your data is consistent across every database and ready for every ServiceNow product. The CSDM provides standard terms and definitions for your CI, and any IT services you use. Map your data to CMDB tables clearly using this modeling to provide it to the apps that use it, across the entire Now Platform.

Dependency Views

Take an in-depth look at your infrastructure and the status of all its components with this plugin. Dependency Views lets you design and customize graphical representations of your CI. Get real-time status updates of your CI, along with data on the applications and services they support, and access to info on any changes, problems, and alerts.

Application services

Use application services to represent and manage operations of various ServiceNow business units and products. An application service is a set of interconnected applications and hosts which are configured to offer a service to the organization such as an organization's internal email system. The various CIs and the relationships between them comprise an application service and are stored in the CMDB.

Data certification

Guarantee your data is safe, stable, and certified for use across the Now Platform and anywhere else it is needed. Data Certification provides options for verifying your data, providing accurate reporting and certification of the condition of all information stored in your CMDB. Define custom parameters for certification, and have data validated with regularly scheduled or on-demand checkups.

Troubleshoot and get help

- [Whitepaper: CMDB Design & Configuration](#)
- [Whitepaper: CMDB Design](#)
- [Whitepaper: Improving Configuration Item Data Quality](#)
- [Whitepaper: CMDB Design Guidance](#)
- [CMDB 101 - What is a configuration management database and why do you need one? \(ServiceNow® Community post\)](#)
- [KB0546686: CMDB Resources Page](#)
- Search the Known Error Portal for known error articles
- Contact Customer Service and Support

Manage the CMDB

With the ServiceNow® Configuration Management Database (CMDB) application, build logical representations of assets, services, and the relationships between them that comprise the infrastructure of your organization. Details about these components are stored in the CMDB which you can use to monitor the infrastructure, helping ensure integrity, stability, and continuous service operation.

Use core features such as CMDB Health, CMDB Identification and Reconciliation, and CMDB CI Lifecycle Management to monitor and detect health issues, reconcile data integrity issues, and manage data life cycle.

Note: CMDB modules, features, and wizards are not supported on mobile devices. You cannot use a mobile device to access the CI Class Manager, Query Builder, or Duplicate CI Remediator. Or to access or configure CMDB features such as Identification and Reconciliation, CMDB Health, CI Lifecycle Management, baseline CMDB, and proposed changes.

[Explore](#)[Set up](#)[Administer](#)

- Upgrade to Vancouver
- Configuration Management and the CMDB
- Domain separation in CMDB

- Populating the CMDB
- CMDB CI Class Models store app

- CI relationships in the CMDB
 - CMDB classifications and class dependency
 - CMDB Identification and Reconciliation
 - CMDB Health
 - CMDB Data Manager
-

Use

- Create a CI class
- Create or edit a CI relationship
- Querying the CMDB
- Apply CMDB remediation

Develop

- CMDB APIs (CMDB SDK)
- Developer training
- Developer documentation

Configuration Management and the CMDB

The Configuration Management data base (CMDB) creates and maintains the logical configurations your network infrastructure needs to support a ServiceNow service.

These logical service configurations are mapped to the physical layout data of the supporting network and application infrastructure in each of your respective domains. They track the physical and logical state of IT service elements and associate incidents to the state of service elements, which helps in analyzing trends and reducing problems and incidents.

The configurations are stored in a configuration management database (ServiceNow CMDB) which consists of entities, called Configuration Items (CI), that are part of your environment. A CI may be:

- A physical entity, such as a computer or router
- A logical entity, such as an instance of a database
- Conceptual, such as a Requisition Service

In each case, there are attributes about the CI that you want to maintain, and there is control you want to have over the CI. There are changes that may need to be made and tracked against the CI. Also, a CI does not exist on its own. CIs have dependencies and relationship with other CIs. For example, the loss of disk drives may take a database instance down, which affects the requisition service that the HR department uses to order equipment for new employees.

It is this relationship data that makes the CMDB a powerful decision support tool. Understanding the dependencies and other relationships among your CIs can tell you, for example, exactly who and what is affected by the loss of that bank of disk drives. When you find out that a router has failed, you will be able to assess the effect of that outage. When you decide to upgrade the processor in a server, you can tell who or what will be affected during the outage.

Configuration items differ from environment to environment because each customer has unique needs. Details about the exact physical attributes of a computer may be needed by one customer, but may represent meaningless data to another. The NOW Platform provides a mechanism to easily define new classes of configuration items and new relationships that may exist between CIs. New classes can be defined that extend other classes. For example, a laptop class exists that extends the computer class. The computer class itself extends the base CI class. Customer class extensions are automatically part of the ServiceNow environment and blend seamlessly into the integration points for other ITIL processes.

You can for example, set the Used for attribute in the cmdb_ci_server table to a value such as 'development', 'test', or 'production'. These values indicate the environment that the CI is supporting, and serve as a way of tracking a CI through its lifecycle in a changing environment.

Extended CMDB

In base systems, CMDB provides core functionality for the configuration management database, including modules for hardware and configuration items. The separate Extended CMDB plugin includes a collection of modules for specialized configuration items, such as radio hardware, test equipment, and voice system hardware.

To extend the CMDB you can [activate](#) the following plugins to access the modules for specialized configuration items.

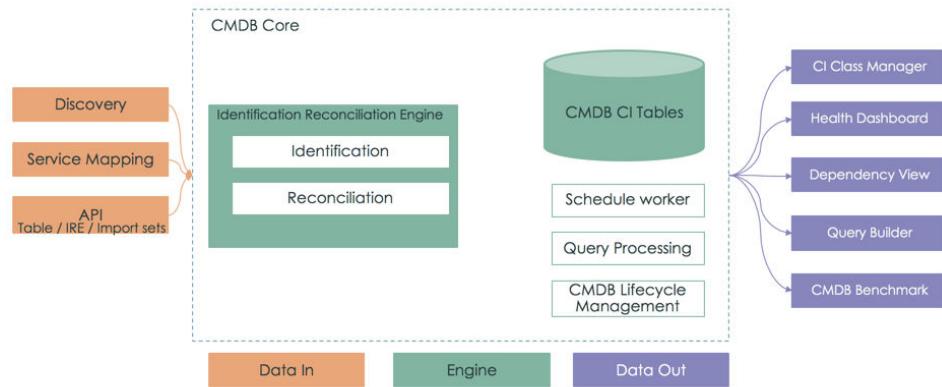
- CMDB Mainframe (com.snc.cmdb.mainframe)
- CMDB Radio Category (com.snc.cmdb.radio.category)
- CMDB Telecom Category (com.snc.cmdb.telecom.category)
- CMDB Test Equipment (com.snc.cmdb.test.equipment)

CMDB hierarchy and CI Class Manager

Sets of CIs that share attributes are stored in their own class table. All CMDB tables are connected by relationships and inherit attributes from each other to form a web of tables referred to as the CMDB hierarchy.

Use the [CI Class Manager](#) to manage CMDB classes within the CMDB hierarchy, CMDB Health, and other class-related definitions. For example, in the CI Class Manager you can view class attributes, class identification rules, and the list of CIs for a specific class. To view the list of CIs in the CMDB, you can also enter `cmdb_ci_list.do` in the filter navigator.

Architecture



Related tables

There are tables that are not part of the CMDB hierarchy but which still qualify as CMDB data. Related tables, such as the Serial Number [cmdb_serial_number] table, don't inherit from the Configuration Item [cmdb_ci] table, but have at least one column that references a CMDB CI. Related tables are specified in the Related Entries [cmdb_related_entry] table.

Some scenarios that involve related tables, can result in orphan or otherwise stale records in related tables. A CI in a related table can, for example, become orphan if the referenced CI in the CMDB is deleted. You can use the [CMDB Data Manager](#) to create a policy of the 'Delete CMDB Related Entry' policy type, that will cascade-delete that un-needed related items data. For more information about creating that CMDB Data Manager policy, see [Create a CMDB Data Manager policy](#).

Roles required

Reading CMDB tables directly requires the cmdb_read role, however accessing the **Configuration** module requires the asset, itil, or itil_admin roles. For viewing CMDB-related records in the user interface, the itil role is usually sufficient. For updating records and for other manipulation of records, roles with higher credentials are usually required, as noted in each procedure throughout the documentation set.

Related concepts

- [CMDB schema model](#)

Domain separation and Configuration Management Database (CMDB)

Domain separation is supported in the Configuration Management Database (CMDB). Domain separation enables you to separate data, processes, and administrative tasks into logical groupings called domains. You can control several aspects of this separation, including which users can see and access data.

Support level: Standard

- Includes **Basic** level support.
- Business logic: The service provider (SP) creates or modifies processes per customer. The use cases reflect proper use of the application by multiple SP customers in a single instance.
- The instance owner must configure the minimum viable product (MVP) business logic and data parameters per tenant as expected for the specific application.

Sample use case: An admin must be able to make comments required when a record closes for one tenant, but not for another.

For more information on support levels, see [Application support for domain separation](#).

Overview

The following topics provide details about domain separation in Configuration Management (CMDB) modules:

- [Domain separation in CMDB Health](#)
- [Domain separation and CMDB Query Builder](#)
- [Domain separation and CMDB Identification and Reconciliation](#)
- [Domain separation and the relations formatter and the CI relationship editor](#)

- CMDB APIs (CMDB SDK)

Related concepts

- Domain separation and Configuration Management Database (CMDB)

CMDB classifications and class dependency

CMDB classifications are groups of configuration items (CIs) that share attributes and are stored in their own class table. Classifications let administrators to define a class hierarchy for the CIs within the CMDB. A CI class refers to the actual table name in the instance database. In that context, 'CI type' is a friendly name that a CI is known by, such as computer, router, or printer.

One of the characteristics of a CMDB class is its dependency on other classes. A class can be independent or dependent, which determines the dependency or independence of the class CIs.

Independent CIs

CIs from an independent class, such as Server CIs, which exist on their own and are not dependent on any other CIs.

Dependent CIs

CIs from a depended class which depend on a relationship to another CI and can't exist on their own in the absence of the dependent relationship. For example:

- Network Adapter CIs can't exist meaningfully without the Hardware CIs that contain them.
- Application CIs can't exist on their own without the Server CI they are hosted on.

To establish CI dependency, you can use the CI Class Manager to specify [dependent relationship rules](#) for a CI class.

For information about creating independent or dependent identification rules to identify independent or dependent CIs respectively, see [Identification rules](#).

- Dependent CIs management

A life cycle update for a CI affects its dependent CIs. For example, when the CI that a dependent CI depends on is deleted, the dependent CI becomes orphan with no further use. To maintain the integrity and health of the CMDB, the system applies cascade-cleanup processes to dependent CIs that are affected by a life cycle update.

- [CMDB record types](#)

The CMDB contains the following major record types.

- [Related Lists of CI components](#)

Related lists in CI records display additional components contained by that CI, such as disk drives on a server and the rules that control the behavior of a network router.

- [Create a CI class](#)

Create a CI class (table) that is an extension of an existing CI class. Then create identification and reconciliation rules for the new class.

- [Reclassify a CI](#)

You can upgrade, downgrade, or switch the class of a CI by modifying its **Class** attribute.

- [Delete CIs](#)

You can use the CI Class Manager to delete CIs that are no longer needed.

- [View and edit class definitions and metadata](#)

Use the CI Class Manager as a central location to explore the CMDB class hierarchy, CI table definitions, and class CIs. View the details of each table such as its label and fields, relationships, and all related metadata definitions.

- [Update the list of classes in the Principal Class filter](#)

Manage the list of classes that are included in the Principal Class filter to restrict the CIs that appear in CIs list views to only specific classes that you need. You can add or remove CMDB classes from the Principal Class filter.

Dependent CIs management

A life cycle update for a CI affects its dependent CIs. For example, when the CI that a dependent CI depends on is deleted, the dependent CI becomes orphan with no further use. To maintain the integrity and health of the CMDB, the system applies cascade-cleanup processes to dependent CIs that are affected by a life cycle update.

For information about independent and dependent CIs, see [CMDB classifications and class dependency](#).

To ensure that dependent CIs are properly managed after deleting or archiving CIs, you must:

1. Enable dependent CIs management as described below.
2. Manually approve the [CMDB Data Manager](#) tasks that dependent CIs management generates, or configure those tasks not to require a review and an approval.

Examples of dependent CIs needing cascade-cleanup:

- Tomcat application (T1) runs on a Linux Server (L1) and contains a WAR file (W1). When L1 is deleted, T1 and W1 become orphan dependent CIs.
- Network Adapters become orphan dependent CIs when the hardware itself is deleted.
- A Linux Server has retired and is set with end of life, the assumption is that if the server is no longer operational then applications which depend on it shouldn't be operational either.

Note: An Orphan dependent CI in the context of Data Manager is different from an orphan CI in the context of CMDB Health. An orphan dependent CI within the context of Data Manager belongs to a dependent class, and is missing the dependent relationship. The definition of an orphan CI in the context of CMDB Health is broader, and includes any CI that matches CMDB Health orphan rules. For more information about orphan CIs in CMDB Health, see [CMDB Health KPIs and metrics](#).

Enable dependent CIs management

To enable dependent CIs management after deleting or archiving CIs:

- Configure your environment for CMDB Data Manager. For details about how to configure the environment for CMDB Data Manager and the CMDB Data Manager own prerequisites, see [CMDB Data Manager](#).
- Ensure that the `cmdb.dependent.ci.cascade.op.enabled` system property is set to **true** (default). This property doesn't exist in the base system and to view or modify the property value, you must first [add it](#) to the [System Properties \[sys_properties\]](#) table.

Cascade-cleanup operations apply only from when you enable the dependent CIs management feature. To apply cascade-cleanup to orphan dependent CIs that already existed in the CMDB before enabling the feature, see [Cascade-cleanup existing orphan dependent CIs](#).

Cascade-retire dependent CIs

When a CI is set to retire, dependent CIs management data processes attempt to cascade-update all the CIs depending on that CI, also to retire.

When a CI is updated to retire (either according to [life-cycle rules](#), or according to CSDM standards in which Life Cycle Stage is **End of Life** and Life Cycle Stage Status is **Retired**), the system checks all the relationship records for that CI. For any relationship that is with a dependent CI, the system adds the dependent CI to the `[cmdb_dependent_ci_ledger]` table. Those CIs are set as being ready to retire using the [CMDB Data Manager](#) upon approval.

Cascade-archive of dependent CIs

Archiving a CI can leave its dependent CIs as orphans in the CMDB. To prevent the accumulation of stale data, the system applies cleanup processes that cascade-archive those orphan dependent CIs.

Orphan dependent CIs are not immediately archived. When a CI is archived, all the CI relationship records for that CI in the `[cmdb_rel_ci]` table, are also archived. The system then checks for any dependent CIs that were orphaned as a result of this archiving. Any CIs in the dependency chain that are identified as orphan dependent CI, are checked for any of the following conditions. CIs that meet any of these

conditions are not orphan dependent CIs and therefore will not be archived:

- The CI is an unhandled duplicate CI (the CI is associated with an unresolved de-duplication task).
- The CI has multiple parent CIs.
- The CI has other relationships in the CI Relationship [cmdb_rel_ci] table.
- The CI belongs to an excluded class. Excluded classes are stored in the CMDB Dependent CI Class Exclusion [cmdb_dependent_ci_class_exclusion] table. In the base system, that table is pre-populated with some classes such as cmdb_ci_vm, cmdb_ci_vmware_instance, and other VMware-related classes. You can manage the set of classes that are exempt from management of orphan dependent CIs, by adding or removing records to that table.

Eventually, only those CIs that these conditions do not apply to are added to the CMDB Dependent CI Ledger [cmdb_dependent_ci_ledger] table. Those CIs are set as being ready for archival using the [CMDB Data Manager](#), upon approval.

Cascade-delete of dependent CIs

Deleting a CI can leave its dependent CIs as orphans in the CMDB. To prevent the accumulation of stale data, the system applies cleanup processes that cascade-delete those orphan dependent CIs.

Orphan dependent CIs are not immediately deleted. When a CI is deleted, all the CI relationship records for that CI in the [cmdb_rel_ci] table, are cascade-deleted. Prior to deleting each of these relationship records, the system checks if the CI on the other end of the relationship belongs to a dependent class. Any CIs in the dependency chain that are identified as dependent on the deleted CI, are checked for any of the following conditions. CIs that meet any of these conditions are not orphan dependent CIs and therefore will not be deleted:

- The CI is an unhandled duplicate CI (the CI is associated with an unresolved de-duplication task).
- The CI has multiple parent CIs.
- The CI has other relationships in the CI Relationship [cmdb_rel_ci] table.

- The CI belongs to an excluded class. Excluded classes are stored in the CMDB Dependent CI Class Exclusion [cmdb_dependent_ci_class_exclusion] table. In the base system, that table is pre-populated with some classes such as cmdb_ci_vm, cmdb_ci_vmware_instance, and other VMware-related classes. You can manage the set of classes that are exempt from management of orphan dependent CIs, by adding or removing records to that table.

Eventually, only those CIs that these conditions do not apply to are added to the CMDB Dependent CI Ledger [cmdb_dependent_ci_ledger] table. Those CIs are set as being ready for deletion using the [CMDB Data Manager](#), upon approval.

Extraneous relationships

For the delete and archive operations, the system also tracks CIs' extraneous relationships in the Dependent CI Relations Evaluation Config [cmdb_dependent_ci_extra_rels_config] table. Extraneous relationships are those relationships that are not in the CI's chain of dependency and deleting or archiving them depends on the value of the cmdb.dependent.ci.extra.rel.check system property. This property is set to **true** by default, in which case extraneous relationships are not deleted or archived. You can set the property to **false** to delete and archive those relationships.

The cmdb.dependent.ci.extra.rel.check property doesn't exist in the base system and to view or modify the property value, you must first [add it to the System Properties \[sys_properties\]](#) table.

Use of CMDB Data Manager to apply cascade-cleanup operations

Dependent CIs management processes use the [CMDB Data Manager](#) to process the life cycle updates for the dependent CIs in the CMDB Dependent CI Ledger [cmdb_dependent_ci_ledger] table in the following ways:

1. Requesting and getting approvals from users for the life cycle updates for the CIs.
2. Performing the actual delete, archive, or retire CI updates after these operations are approved.

The system generates 'Dependent CI - Deletion', 'Dependent CI - Archive', and 'Dependent CI - Retire' Data Manager policies, for the set of CIs that are ready to be deleted, archived, or retired. These policies

are set with the respective 'On Demand' policy types. Then, a Data Manager admin or user, according to the Managed By Group setting, must review and approve these tasks before the Data Manager applies the delete, archive, or retire subflows to the dependent CIs.

To automatically approve those tasks, use the CMDB Data Manager to clear the **Needs Review** flag of the respective Data Manager policies. Those tasks will then run without requiring any user intervention.

Cascade-cleanup existing orphan dependent CIs

When enabling the dependent CIs management feature, the cascade-cleanup operations apply only from when the feature is enabled. However, it might be necessary to apply a similar cascade-cleanup operation to orphan dependent CIs that already existed in the CMDB before the feature was enabled.

[Activate](#) the Cleanup Orphan CIs scheduled job to perform a one-time cascade-cleanup of orphan dependent CIs across the CMDB. The Cleanup Orphan CIs scheduled job checks throughout the entire CMDB to identify any orphan dependent CIs. These CIs are then processed in the same manner that CIs that are cascade-deleted are processed. CIs that are ready to be deleted are added to the CMDB Dependent CI Ledger [cmdb_dependent_ci_ledger] table. The CMDB Data Manager is then leveraged as described in the [Use of CMDB Data Manager to apply cascade-cleanup operations](#) section.

The Cleanup Orphan CIs scheduled job is intended to run only once, after which the job deactivates itself. Depending on the size of the CMDB, it might take the Cleanup Orphan CIs scheduled job several days to complete.

Related tasks

- [Create a CI class](#)
- [Reclassify a CI](#)
- [Delete CIs](#)
- [View and edit class definitions and metadata](#)
- [Update the list of classes in the Principal Class filter](#)

Related reference

- [CMDB record types](#)
- [Related Lists of CI components](#)

CMDB record types

The CMDB contains the following major record types.

CMDB record types

Record types	Description
Configuration Item (CI)	Any computer, device, or service in the CMDB. A CI's record includes all of the relevant data, such as manufacturer, vendor, location, etc. Configuration items can be created or maintained either using tables, lists, and forms within the platform, or using the Discovery application.
Relation Type	A defined relationship between a CI and either another CI, a user, or a group. Relation types are defined twice, once from the perspective of the child CI and once from the parent CI's perspective. For example, a parent CI that powers a child CI uses relation type Powers::Is Powered By . Example relation types include In Rack::Rack contains , Log Reviewed by::Reviews logs for, or Backup done by::Does backups for. CMDB relationships can be established using Discovery or

Record types	Description
	using the tables, lists, and forms within the platform. The CMDB form has a specific Related Items toolbar optimized for modifying relationships.

Related tasks

- [Create a CI class](#)
- [Reclassify a CI](#)
- [Delete CIs](#)
- [View and edit class definitions and metadata](#)
- [Update the list of classes in the Principal Class filter](#)

Related concepts

- [Dependent CIs management](#)

Related reference

- [Related Lists of CI components](#)

Related Lists of CI components

Related lists in CI records display additional components contained by that CI, such as disk drives on a server and the rules that control the behavior of a network router.

When Discovery runs, the Related List is populated with the components that Discovery finds running on the CI. The CI record might show different lists from scan to scan, depending on whether or not Discovery found the component.

By default, the Related Lists only display those components that are associated with that CI in the CMDB that has been discovered by the

last scan. Components that are recorded in the CMDB but are not discovered in a scan, are deemed absent and do not appear in the list.

There are two types of components that appear in the Related List: components that are CIs themselves (such as hard disks), and components that are not (serial numbers and rules). The default filter condition in the breadcrumbs for components that are CIs is **[Status] [=] [Absent]**. The filter condition for components that are not CIs is **[Absent] [=] [false]**.

For example, a router can have several Related Lists affected by these filter conditions, including routing rules, disk drives, interfaces, and network adapters. Only those components found during the last Discovery appear in these Related Lists.

- [Teams related list](#)

The Teams related list associates a user group to a CI based on group type, providing flexibility in tracking the different types of groups assigned to a CI. The Teams related list appears on CI forms for CIs of the Service [cmdb_ci_service] class and its descendent classes such as the Application Service [cmdb_ci_service_auto] class.

Related tasks

- [Create a CI class](#)
- [Reclassify a CI](#)
- [Delete CIs](#)
- [View and edit class definitions and metadata](#)
- [Update the list of classes in the Principal Class filter](#)

Related concepts

- [Dependent CIs management](#)

Related reference

- [CMDB record types](#)

The Teams related list associates a user group to a CI based on group type, providing flexibility in tracking the different types of groups assigned to a CI. The Teams related list appears on CI forms for CIs of the Service [cmdb_ci_service] class and its descendent classes such as the Application Service [cmdb_ci_service_auto] class.

Using the Teams related list can be useful if many data sources are used in the organization, and when using the [IntegrationHub ETL](#).

In the base system, the Teams related list contains group types that match the fields:

- Approval Group
- Change Group
- Managed by Group
- Support Group

When you set a group assignment on a CI form of one of those classes, that group assignment is automatically synchronized with the Teams related list. If you set an assignment group for an application service, a relationship record is created to represent the new group assignment for the CI. The Teams related list on such CI forms, always shows the current settings for the various group assignments for the CI.

If there are multiple groups assigned to a CI, then on the **Teams** related list on a CI form, you can designate one of those groups as primary. When an incident involving that CI is created, the incident is assigned to that primary group. Only one group type can be designated as primary for a CI at any given moment.

As an example, use the Teams related list for an application service with the following characteristics:

- Requires access to a database
- Runs on Linux servers

- Has a group assigned to manage the software portion of the actual application

In that example, you can track all of these group assignments by adding the appropriate group types and setting CIs with the new custom group types.

By-directional synchronization

When you set or modify the value of one of the assignment group fields on a Service CI form, then the Teams related list is updated with that change, adding a Teams related record if a corresponding one doesn't exist. For example, when you set an empty **Support group** field to **Database San Diego** and save the form, then the system adds a Teams record in which Group type is **Support Group** and **User group** is **Database San Diego**.

In the opposite direction, any change in a Teams related record, where Group type corresponds one of the group assignment fields in the CI, is synchronized to the corresponding field. For example, if you set User group to **CAB Approval** in the Support Group group type, the **Support group** field on the CI form is updated with the **CAB Approval** value.

Deleting a Teams record doesn't affect the group assignment fields in the CI.

Add the Teams related list to a Service CI form

By default, the tab for the Teams related list doesn't appear in the Related Lists section on Service CI forms. You can add the Teams related list by selecting the form's Additional actions menu, selecting **Configure** and then **Related Lists**. In the Available list that appears, move **Teams** to the Selected list and then select **Save**.

Add a custom group type to the Teams related list

The Teams related list lets you add custom group types that are needed in your organization, extending the initial list of supported group types.

Before you begin

Role required: itil_admin, asset, or cmdb_admin

About this task

To add a group type, you must modify the dictionary definition of the group_type column in the cmdb_rel_team table.

Procedure

1. On a Service CI form, select the **Teams** related list.
2. In the **Columns** tab, select the **Group type** column.
3. On the Dictionary Entry form, in the **Choices** related list tab, insert a new row for a new Teams choice such as **Datacenter Group**.
4. Save your changes.

Create a CI class

Create a CI class (table) that is an extension of an existing CI class. Then create identification and reconciliation rules for the new class.

Before you begin

The class that is being extended must have its Is_Extendable field checked, indicating that the class is extendable.

Role required:

- Itil_admin and personalize_dictionary: Required for editing the dictionary table
- admin: Full access

About this task

The CI Class Manager is a centralized location for managing CMDB tables and for creating a class that is derived from another CMDB class. Creating a class requires basic details such as a label and a name. Identification and reconciliation rules are also required to ensure that the class can be successfully identified by the identification engine.

For more information about extending a class and how attributes are derived from a parent class in that process, see [Table extension and classes](#).

Procedure

1. Navigate to **All > Configuration > CI Class Manager**.
2. Click **Hierarchy** to expand the CI Classes list.
3. Select the class that the new class is extended from.
4. Click **Add Child Class**.
The **Add Child Class** option appears only if the selected class is extendable.
5. On the **Provide Basic Info** tab, fill out the information and then click **Next**.

Field	Description
Display name	A unique label for the class (such as Laptops or Thin Clients). The label appears on list and form views for the class. Updating the Label field also updates the label record in the language file for the current language. See Field Labels in Data dictionary tables . Maximum string length is 80 characters.
Table name	Automatically populated based on the table label and the prefix string 'u_cmdb_ci'. You cannot modify the prefix; however, you can modify the rest of the table name. The name can contain only

Field	Description
	lowercase, alphanumeric ASCII characters and underscores (_). Maximum string length is 80 characters.
Description	Explanation of the use purpose of the class.
Icon	The icon associated with the class.
Extensible	Indicator of whether this class can be extended.
Principal Class	Denotes whether this class is included in the Principal Class filter . If this class is included in the Principal Class filter, then CIs from this class appear in CI list views when the Principal Class filter is applied.

6. On the **Add Attributes** tab, click the + sign and enter details for each new class column.
7. Click **Next**.
For description of the different columns in the list view, see [Dictionary entry form](#). Double-click the value in the Identification Rule column and set it to true to designate the column as a CI identifier for class identification.
8. On the **Set Identification Rule** tab, examine the **Derived** identification rule and its **Identifier Entries**.
You can click **Replace** to replace the derived rule with a new identification rule and new identifier entries specific to the new class. See [Identification rules](#) for details about identification rules and identifier entries.
9. On the **Dependencies** tab, click **Add dependency** to add dependent rules.

The **Dependencies** tab appears only if there are dependent identification rules for the selected class.

10. On the **Add Reconciliation Rules** tab, click **Add** to create the following rules:
 - a. [Reconciliation Rules](#)
 - b. [Data Refresh Rules](#)
11. On the **Add Suggested Relationships** tab, review the diagram of the class derived suggested relationships.
12. Use the filter to display only inbound, outbound, or specific relationship types.
13. To add a suggested relationship for the class:
 - a. Click **New**.
 - b. In the Add Suggested Relationship dialog box, select a **Relationship** and a **Target Class** for the relationship. **This Class** and the **Target Class** become parent or child in the suggested relationship, based on your selection of the **Relationship**.
 - c. Click **Save**.When building relationships for the class in the Query Builder, the list of suggested relationships is updated.
14. Click **Done**.

Related tasks

- [Reclassify a CI](#)
- [Delete CIs](#)
- [View and edit class definitions and metadata](#)
- [Update the list of classes in the Principal Class filter](#)

Related concepts

- [Dependent CIs management](#)

- CMDB Identification and Reconciliation

Related reference

- CMDB record types
- Related Lists of CI components

Reclassify a CI

You can upgrade, downgrade, or switch the class of a CI by modifying its **Class** attribute.

Before you begin

Role required: itil or asset (In general, the roles required to update a CI)

About this task

Each class is defined with a unique set of attributes. This set consists of attributes that were derived from the parent class, and additional attributes defined for the class.

When you reclassify a CI, the following occurs.

1. The set of attributes is adjusted to match the set of attributes of the newly assigned class. Attributes are added or removed as needed.
2. If any attributes are unique to the current class and are not defined in the newly reclassified class, they are lost.
3. A new record with the CI's current sys_id is inserted to the table of the new class, with the appropriate set of attributes for the class (the sys_id of the CI is retained).

Depending on the reclassification, the following occurs.

Downgrade

The CI class is updated to a class that is higher in the class hierarchy, and the newly assigned class is a parent of the current class. For example, reclassifying a CI from the cmdb_ci_server class to the cmdb_ci_computer class.

For example, the cmdb_ci_server class has attributes that the cmdb_ci_computer class does not have. During the downgrade, these attributes and their respective values are not included in the new CI record that is inserted into the cmdb_ci_computer class.

Upgrade

The CI class is updated to a class that is lower in the class hierarchy, and the newly assigned class is a derived child of the current class and has additional attributes. For example, reclassifying a CI from the cmdb_ci_computer class to the cmdb_ci_server.

Switch

The newly assigned class is in a different branch in the class hierarchy and has a different set of attributes than the current class. For example, reclassifying a CI from the cmdb_ci_linux_server class to the cmdb_ci_win_server class.

A switch is a combination of a downgrade and an upgrade. For example if the CI is downgraded to the cmdb_ci_server, and then upgraded to the cmdb_ci_win_server class. Therefore, attributes are lost in the same manner as in a downgrade operation.

Note: Avoid the CI class downgrade and CI class switch operations as those can lead to data loss. When automatic CI reclassification is enabled (which is by default), the [identification process](#) can result in some automatic reclassifications which lead to data loss.

For information about CI reclassification by the Identification and Reconciliation Engine (IRE) and related system properties that control the behavior of automatic CI reclassification, see [CI reclassification during IRE processing](#).

Procedure

1. Locate the CI that you want to reclassify and display it in a list view. You can use the application navigator. Or for example, if the CI is a server, then in the navigation search box, type `cmdb_ci_server.list` to display the CI in the **Servers** view.
2. Ensure that the **Class** field is displayed in the list.

If you do not see this attribute, personalize the list to add the **Class** field.

3. Double-click the **Class** value for the CI, and select a new class.
4. Click the green check box to confirm your selection.

Related tasks

- [Create a CI class](#)
- [Delete CIs](#)
- [View and edit class definitions and metadata](#)
- [Update the list of classes in the Principal Class filter](#)

Related concepts

- [Dependent CIs management](#)

Related reference

- [CMDB record types](#)
- [Related Lists of CI components](#)

Delete CIs

You can use the CI Class Manager to delete CIs that are no longer needed.

Before you begin

Role required: itil_admin

About this task

For information about policy-based, large scale automated CI deletions, see [CMDB Data Manager](#).

Note: You can't delete base system tables. For information about deleting custom tables, see [Delete a table](#).

Procedure

1. Navigate to **All > Configuration > CI Class Manager**.
2. Click **Hierarchy** to expand the CI Classes list and then select the class from which you want to delete CI records.
3. In the class navigation bar on the left, click **CI List**.
4. On the CI List form view, select the CIs that you want to delete. Select the check box in the header to select all the CIs that are visible.
5. Click **Actions on selected rows** and then click **Delete**.
6. Click **Delete** in the **Confirmation** dialog box.

Result

After deleting CIs that a dependent CI depends on, the dependent relationship is also deleted. The dependent CI becomes an orphan and is not immediately deleted. In this situation, the system attempts to cascade delete such orphan dependent CIs to prevent the accumulation of stale data and maintain the health of the CMDB. For information about how the system manages orphan dependent CIs, see [Management of orphan dependent CIs](#).

Related tasks

- [Create a CI class](#)
- [Reclassify a CI](#)
- [View and edit class definitions and metadata](#)
- [Update the list of classes in the Principal Class filter](#)

Related concepts

- [Dependent Cls management](#)

Related reference

- [CMDB record types](#)
- [Related Lists of CI components](#)

View and edit class definitions and metadata

Use the CI Class Manager as a central location to explore the CMDB class hierarchy, CI table definitions, and class Cls. View the details of each table such as its label and fields, relationships, and all related metadata definitions.

Before you begin

Role required: none

About this task

The CI Class Manager displays the entire CMDB class hierarchy in a tree-view format, consolidating class definitions into a central location. It lets you display metadata information for a class, such as reconciliation rules, mandatory and recommended fields, and audit templates. You can also select a specific class to view, to modify, or to extend its definition to create a derived class. For each class, you can directly access CMDB Health settings, identification and reconciliation rules, orphan scorecard, and certificate template, defined for the class.

For more information about extending a class and how attributes are derived from a parent class in that process, see [Table extension and classes](#).

Procedure

1. Navigate to **All > Configuration > CI Class Manager**.
2. Click **Hierarchy** to expand the CI Classes list and then select a class to display details for.

3. On the class navigation bar, expand the following items to display further details for the class.

- **Class Info:**

- **Basic Info:** Displays details for the selected class, such as the display and table name, description, and class icon. Lets you edit some of the class definitions, and prevents editing of some details such as the table name.

Role required: itil for reading, and itil_admin and personalize_dictionary for writing.

- **Attributes:** Displays table attributes (columns). Lets you edit those attributes and add new ones. For description of the different attributes in the list view, see [Dictionary entry form](#).

Role required: personalize_dictionary for reading and writing

To add an attribute:

- a. Click the **Added** tab and scroll to the bottom of the list.
 - b. Double-click **Insert new column**, and enter details for each new class attribute. Set **Identification** to true to designate an attribute as a CI identifier for class identification.
 - c. Click **Save**, and fix any errors that appear.
- **Identification and Reconciliation:** Displays and lets you edit, create, and delete identification and [inclusion rules](#), reconciliation and [data refresh rules](#) for the class.

See [CMDB Identification and Reconciliation](#) for more information.

Role required: itil for reading, and itil_admin (on top of itil) for writing.

- **Dependent Relationships:** Displays and lets you edit, create, and delete hosting and containment relationships for the class. See [CMDB dependent relationship rules](#) for more information.

Role required: itil for reading and itil_admin (on top of itil) for writing.

- **Suggested Relationships:** Displays a diagram of all suggested relationships for the class, and lets you delete or add suggested relationships for the class. Use the navigation tools to increase or decrease the diagram, and to move the diagram on the page. Use the filter to display specific relationship types. See [Suggested class relationships](#) for more information.

Role required: itil.

- **All Relationship Rules:** Displays a combined diagram of all suggested relationships and all dependent relationships for the class. Use the navigation tools to zoom in or out, and to move or center the diagram on the page. Use the filter to display specific relationship categories.
- **Health:** Lets you review and configure CMDB Health-related system properties, scorecards, and rules and settings for all CMDB health KPI and metrics, at the class level. See [CMDB Health](#) for information about enabling and configuring CMDB Health, and displaying health reports.

Role required: Itil for reading and itil_admin (on top of itil) for writing.

- **CI List:** Displays the CIs of the selected class. Lets you create CIs of the selected class and perform other operations such as delete.

Role required: Itil for reading. Writing requirements follow the selected table settings.

Related tasks

- [Create a CI class](#)
- [Reclassify a CI](#)
- [Delete CIs](#)
- [Update the list of classes in the Principal Class filter](#)

- [Create or modify map icons](#)

Related concepts

- [Dependent CIs management](#)

Related reference

- [CMDB record types](#)
- [Related Lists of CI components](#)

Update the list of classes in the Principal Class filter

Manage the list of classes that are included in the Principal Class filter to restrict the CIs that appear in CIs list views to only specific classes that you need. You can add or remove CMDB classes from the Principal Class filter.

Before you begin

Role required: `itil_admin` and `personalize_dictionary`

About this task

Apply the CMDB Principal Class filter in list views of CMDB CIs so that only CIs that belong to the classes in the filter, appear. In a base system, the Principal Class filter doesn't contain any classes. The principal class setting applies only to the current class and is not derived by child classes.

For more information about list view filters, see [Save and use filters in a list view](#).

Procedure

1. Navigate to **All > Configuration > CI Class Manager**.
2. Click **Hierarchy** to expand the CI Classes list and then select a class to add or remove from the Principal Class filter.
3. On the class navigation bar, navigate to **Class Info > Basic Info**.

4. On the Basic Info form, select or clear **Principal Class**.

5. Click **Save**.

Result

The Principal Class filter is updated with the addition or the removal of the class from the list of classes in the filter. When you apply the Principal Class filter to a Configuration Items list view, only CIs from classes included in the filter, appear.

What to do next

In both of the following scenarios, the list of CIs refreshes to display only CIs whose class is included in the Principal Class filter:

- 1. In the **Filter navigator**, type `cmdb_ci.list` and then press the Enter key.
 2. In the Configuration Items list view, click the **List controls** menu icon, select **Filters** and then click **Principal Class**.
- 1. Open a Change Request form.
 2. Scroll down and select the **Affected CIs** tab. Click **Add**.
 3. In the **Add Affected CIs** form, click the **List controls** menu icon, select **Filters** and then click **Principal Class**.

For more information about adding affected CIs to change requests, see [Associated CIs on a change request](#).

Related tasks

- [Create a CI class](#)
- [Reclassify a CI](#)
- [Delete CIs](#)

- View and edit class definitions and metadata

Related concepts

- [Dependent Cls management](#)

Related reference

- [CMDB record types](#)
- [Related Lists of CI components](#)

CI relationships in the CMDB

The CMDB, in contrast to a static asset list, helps you track not only the configuration items (Clis) within your system, but also the relationships between those items.

A relationship in the CMDB consists of two Cls and a relationship type:

- Parent Cl
- Child Cl
- Type of the relationship that links both Cls

For example, in the [Server1] [Managed by] [Server2] relationship:

- Server1 is the child Cl
- Server2 is the parent Cl
- [Managed by] is the relationship type

For example, a web application might read data from an instance of Oracle, which in turn might depend on a piece of underlying hardware. Most Cls in a CMDB have multiple relationships to other Cls, users, and groups.

The relationships between Cls can be automatically discovered. If you use Discovery, many relationships can be automatically loaded into the system through the discovery process. If you import your data from another system, you get some form of relationships.

You can add to automatically discovered relationships, create relationships, or edit relationships for a CI by launching the [CI relationship editor](#) from the CI form.

Dependent and non-dependent relationships

[Dependent relationships](#), such as tomcat RunsOn Hardware, are used by the Identification and Reconciliation Engine (IRE) to identify dependent CIs.

Non-dependent relationships are not used for CI identification, and therefore can be deleted if no longer needed. CMDB tracks discovery source and last scanned time for non-dependent relationship in the Relationship Sources [sys_rel_source] table. Dependent relationships are used for CI identification. Therefore they should not be directly deleted and they are not tracked.

Information in the Relationship Sources [sys_rel_source] table can be used to decide if it is safe to delete a non-dependent relationship. For example, a discovery source which is attempting to delete a non-dependent relationship can confirm that:

- There are no other data sources for that relationship.
- The relationship was not updated for some specified length of time and therefore is no longer needed.

When a non-dependent relationship is deleted from the CI Relationship [cmdb_rel_ci] table, all cascading corresponding records in the Relationship Sources [sys_rel_source] table are deleted.

Key relationships

The following table contains descriptions for some key CMDB relationships.

Parent	Child	Description
Applicative Flow To	Applicative Flow From	Connections between endpoint CIs.

Parent	Child	Description
		<p>Note: For internal use only (service model).</p>
Connects to	Connected by	<p>Network Connections between elements that are talking to each other.</p> <p>Examples: Workstation to switch, switch to switch, kubernetes workload to service.</p>
Contains	Contained by	<p>Typically a containment relationship (CI to contained CI). The child CI typically has a single parent CI with this relationship type.</p> <p>Examples: Tomcat to Tomcat WAR, VMware Datacenter contains Network.</p>
Defines resources for	Gets resources from	<p>Parent CI defines/gets resources from a child CI.</p> <p>Example: VMware - Resource pool gets</p>

Parent	Child	Description
		resources from ESX Server.
Depends on	Used by	Parent CI depends on child CI. Meaning that problem/change in the child CI may impact the parent CI.
Hosted on	Hosts	<p>Hosting relationship between an element and its host.</p> <p>Examples: Cloud resource to logical data center, k8s workload to k8s cluster.</p>
Implement End Point To	Implement End Point From	<p>Endpoint to CI that exposes this endpoint.</p> <p>Note: For internal use only (service model).</p>
Manages	Managed by	<p>Typically used where one CI manages one or more other CIs.</p> <p>Example: vCenter manages vCenter Datacenter.</p>

Parent	Child	Description
Members	Member of	<p>Typically used with clusters where a cluster node is a member of a cluster.</p> <p>Example: ESXi Server is a member of vCenter Cluster.</p>
Owns	Owned by	<p>Usually a containment relationship (CI to owned CI). The child CI typically has a single parent with this relationship type.</p>
Runs on	Runs	<p>Typically between a CI that represents a software application, to the hosting hardware/VM.</p> <p>Example: Tomcat 'Runs on' Linux server.</p>
Use End Point To	Use End Point From	<p>From the CI to an outgoing endpoint.</p> <p>Note: For internal use only (service model).</p>

- Suggested class relationships

The system keeps a table (Suggested Relationship [cmdb_rel_type_suggest]) of relationship types that are appropriate for a CI type, based on its class. You can manage suggested relationships by navigating to **Configuration > Suggested Relationships**, or in the CI Class Manager.

- [Add a suggested relationship](#)

Add a suggested relationship for a class. The list of suggested relationships for a class is available when you create a new relationship for a CI of that class.

- [Relationship governance rules](#)

Relationship governance rules is a set of relationship rules used to ensure consistency and validity in modeling relationships between configuration items (CIs) in the CMDB. Use relationship governance rules to prevent the selection of relationship types or directions that are not allowed between specific CI types.

- [CI relations formatter](#)

The default CI form includes a CI relations formatter from which you can examine a CI and its relationships in various views. From the CI relations formatter, you can also launch the CI relationship editor for the CI.

- [CI relationship editor](#)

Use the relationship editor to view, create, modify, or delete CI relationships. Open the relationship editor from the CI Relations formatter.

- [Relation qualifier](#)

A relation qualifier, which is a CI of the Qualifier [cmdb_ci_qualifier] type, stores important information about the CI relationships.

- [CI relationship security](#)

When applying security to CI relationships, it is important to apply the access controls both to the CI Relationship (cmdb_rel_ci) table and to create an operation editCIRelations to the * table as well.

- [Create a CI relation rollup](#)

A CI relation rollup allows you to sum, count, max, min, or mean a relationship type. You can create CI relation rollups.

Related tasks

- [Create a CI relation rollup](#)

Suggested class relationships

The system keeps a table (Suggested Relationship [cmdb_rel_type_suggest]) of relationship types that are appropriate for a CI type, based on its class. You can manage suggested relationships by navigating to **Configuration > Suggested Relationships**, or in the CI Class Manager.

Suggestion model

The relationship editor has a base CI. The base CI designates the CI that a user was on before launching the editor, as the base CI in the new relationship. If you launched the relationship editor from the Inux100 CI, then Inux100 becomes the base CI. Also, every CI in the system has a type (class). For example, bond Inux100 is of the Linux server type.

Many CI types are children of other types in the hierarchy. For example, the class hierarchy for a Linux server is:

```
cmdb_ci -> cmdb_ci_hardware ->cmdb_ci_computer ->  
cmdb_ci_server -> cmdb_ci_linux_server
```

The suggestion model works by analyzing the suggested relationship table for all relationships whose base class is the current base class of the user or any one of its parent classes. For example, looking at a Linux server, the suggestion model would retrieve any relationships whose base class was:

```
cmdb_ci_linux_server, cmdb_ci_server, cmdb_ci_computer,  
cmdb_ci_hardware, or cmdb_ci
```

ITOM Visibility, if available, uses enhanced discovery patterns to identify and add CI relationships to the Suggested Relationships table.

Suggested CI relationships in the relationship editor

The CI relationship editor uses the suggestion model to help users select reasonable relationships for configuration items.

For example, consider these relationship types in the system:

- Provides Power for :: Receives Power From
- Runs on :: Hosts

Typically, a user uses these relationships to define the following reasonable relationships between two items as follows:

- a database runs on a server
- a rack provides power for a server

Typically, neither of the following definitions would be appropriate:

- a rack runs on a server
- a server runs on a database

For descriptions of some key relationships, see [CI relationships in the CMDB](#).

Related tasks

- [Add a suggested relationship](#)
- [Create a CI relation rollup](#)
- [CI Class Manager](#)

Related concepts

- [Relationship governance rules](#)
- [CI relations formatter](#)
- [CI relationship editor](#)
- [Relation qualifier](#)

- CI relationship security

Add a suggested relationship

Add a suggested relationship for a class. The list of suggested relationships for a class is available when you create a new relationship for a CI of that class.

Before you begin

Role required: To view — itil. To create, update, or delete suggested relationships — itil_admin.

Procedure

1. Use the CI Class Manager (Role required: itil_admin):
 - a. Navigate to **All > Configuration > CI Class Manager**.
 - b. Click **Hierarchy** to expand the CI Classes list. Then select the class to add a suggested relationship to.
 - c. In the class navigation bar, click **Suggested Relationships**.
 - d. Click **New**.
 - e. In the Add Suggested Relationship dialog box, select a **Relationship** and a **Target Class** for the relationship. **This Class** and the **Target Class** become parent or child in the suggested relationship, based on your selection of the **Relationship**.
 - f. Click **Save**.
2. Or, navigate to **All > Configuration > Relationships > Suggested Relationships** (Role required: admin):
 - a. Click **New**.
 - b. Complete the form.

Suggested Relationship fields

Field	Description
Base class	The base class in the relationship, which depending on the relationship type, is either the parent or the child in the relationship.
Relationship	Relationship type.
Dependent class	The dependent class in the relationship, which depending on the relationship type, is either the parent or the child in the relationship.

Example: Suggested relationship you can add

Base Class	Relationship	Dependent/Target Class
Oracle	Is Hosted On	Linux Server
Oracle	Is Hosted On	Solaris Server

Note: The same parent class and relationship can appear more than once.

What to do next

You may need to delete a suggested relationship, for example, to limit the choice of available relationships in the CI relationship editor. Removing a suggested relationship does not affect relationships that are created or updated by Discovery.

Related tasks

- [Create a CI relation rollup](#)

Related concepts

- [Relationship governance rules](#)
- [CI relations formatter](#)
- [CI relationship editor](#)
- [Relation qualifier](#)
- [CI relationship security](#)

Related reference

- [Suggested class relationships](#)

Relationship governance rules

Relationship governance rules is a set of relationship rules used to ensure consistency and validity in modeling relationships between configuration items (CIs) in the CMDB. Use relationship governance rules to prevent the selection of relationship types or directions that are not allowed between specific CI types.

Different applications such as Discovery and Service Mapping, create relationships between CIs. Each application might use inconsistent relationship type or direction to represent the same entity, resulting in multiple views of the same CIs. Relationship governance rules define what are valid relationship types and valid directions between pairs of CI types resulting in valid and consistent relationships in the CMDB.

Relationship governance rules consist of:

- **CMDB dependent relationship rules:** Rules (hosting and containment rules) that are used for CI identification. You can view and modify dependent relationship rules in the CI Class Manager, after selecting a class from the class hierarchy and clicking **Dependent Relationship**.
- **Suggested relationships:** Rules that are based on existing suggested relationships in the Suggested Relationship [cmdb_rel_type_suggest] table. Suggested relationships are used in the [CI relationship editor](#). You can view and modify suggested relationships in the CI Class Manager,

after selecting a class from the class hierarchy and clicking **Suggested Relationships**.

- **Reference rules:** Rules that are used mostly by Cloud Management to represent all the possible valid combinations of pairs of referencing and referenced CIs in the service definition.
- Built-in valid relationships: The following relationships are pre-defined in the base system as valid relationships:
 - cmdb_ci_endpoint -> Applicative Flow To::Applicative Flow From -> cmdb_ci_endpoint
 - cmdb_ci_endpoint -> Implement End Point To::Implement End Point From -> cmdb_ci
 - cmdb_ci -> Use End Point To::Use End Point From -> cmdb_ci_endpoint

General behavior

- Relationship governance rules support inheritance.

For example, suppose that the suggested relationship cmdb_ci_appl Runs On::Runs cmdb_ci_hardware exists. Then a Runs On::Runs relationship between a cmdb_ci_appl_dot_net CI and a cmdb_ci_windows_server CI is valid. That is because .Net Application class inherits from the Application class and the Windows Server class inherits from the Hardware class.

- Duplicate relationship governance rules are not allowed.
- Relationship governance rules are not domain separated.
- It is allowed to have more than one relationship type between the same two CI types.

For example, the following relationships are valid:

- cmdb_ci_appl Depends On::Used by cmdb_ci_service
- cmdb_ci_appl Receives data from::Sends data to cmdb_ci_service

Reports

A relationship between CIs is considered valid if it conforms to any of the relationship governance rules. Use the CMDB dashboard to view reports about [overall relationships health](#) including relationships compliance with relationship governance rules. The 'Relationships not compliant with all relationship rules' report shows CI relationships that are not compliant with any of the relationship governance rules.

Related tasks

- [Add a suggested relationship](#)
- [Create a CI relation rollup](#)

Related concepts

- [CI relations formatter](#)
- [CI relationship editor](#)
- [Relation qualifier](#)
- [CI relationship security](#)

Related reference

- [Suggested class relationships](#)

CI relations formatter

The default CI form includes a CI relations formatter from which you can examine a CI and its relationships in various views. From the CI relations formatter, you can also launch the CI relationship editor for the CI.

If the domain separation plugin is activated, then only relationships in which the logged on user is authorized to view both CIs, are displayed.

The CI relations formatter contains a list of related CIs and a toolbar with controls for viewing the relationships between the current CI and related CIs. You can configure the controls in this formatter to modify varying

aspects of the view. For more information about formatters, see [Create a formatter and add it to the form](#).

Note:

- If an endpoint is a child in one relationship and the same endpoint is a parent in another relationship, then that endpoint is hidden and does not appear in the relations formatter view. Similarly, relationship qualifier chains are also hidden and do not appear in the relationship formatter view.
 - Example: CI1 > endpoint > CI2

In this example, CI1 is related to CI2 through relationships with endpoint. A single relationship appears in the relations formatter:

CI1 > CI2 (These relationships appear as a direct relationship without endpoint, because endpoint is a parent in one relationship and a child in another relationship).

- Example: CI1 > endpoint1 > CI2 > endpoint2

Two relationships appear in the relations formatter:

CI1 > CI2 (endpoint1 is hidden because it is a parent in one relationship and a child in another relationship).

CI1 > CI2 > endpoint2 (appears as level 2 relationship – endpoint1 is hidden and endpoint2 appears as it a child and not a parent in any other relationship).

- On instances that do not meet the internet browser requirements for the CI relations formatter, the default CI form includes the legacy CI relations formatter instead. For more information, see [Legacy CI relations formatter](#).
- CIs not extended from the Configuration Item [cmdb_ci] table, are not displayed in Dependency Views maps and in CI relation formatters.
- The Applicative Flow To::Application Flow From relationship is a special relationship type used only between Service Mapping endpoints. This relationship type is not intended for use in the CMDB as a relationship between CIs and therefore it is not displayed in the relations formatter.

Controls for viewing related CIs

Control	Definition
 Add CI relationship	Starts the relationship editor to manually create CI relationships. For more information, see Create or edit a CI relationship .
 Show dependency views	Launches a Dependency Views map in another window or tab. The CI is the central node in the map, with a configurable number of levels above and below that node in the hierarchy. Map indicators next to the nodes indicate the number of tasks, incidents, problems, changes, or outages related to that node. Right-click to expand collapsed nodes or display a list of related tasks or problems. For more information, see Dependency Views map .
Search for CI	Filters the CIs included in the display.



Click the **Settings** () icon to configure additional view settings that filter the data displayed. Settings are preserved through logging out and logging back in.

Related items settings

Setting	Description
Show Relations in Flat/Tree Layout	To view a flat list of related CIs that are grouped by relationship

Setting	Description
	<p>type in alphabetical order, click Flat (default value).</p> <p>To view groups of related CIs in a hierarchical tree, click Tree. If you select the tree view, you cannot configure any other settings for viewing related CIs. A single list of upstream and downstream relationships is displayed.</p>
Show Relations in Split/Merge Layout	<p>To view a single list that includes both upstream and downstream relationships, click Merge (default value). Relationships are grouped by relationship type.</p> <p>To view separate lists for upstream and downstream relationships, click Split.</p>
Filter Relations by Max Level	<p>Select the number of downstream and upstream levels in the hierarchy to include when displaying CIs in a flat view.</p> <p>Default value is 3.</p>
Filter Relations by Relationship Type	<p>Select the types of relationships to view.</p> <p>Default value is 'All Relationship Types'.</p>

Setting	Description
Filter Relations by CMDB View	Filter by tables specified in CMDB views, if any relationship filters exist.

The relations formatter uses the following icons to provide additional information about changes, problems, and outages related to CIs in the relationship:

Icons related to CIs

Icon	Description
	Recently closed changes
	Planned changes
	Currently open changes
	Recently closed outages
	Problems
	Incidents
	Planned outages
	Currently open outages

In large networks, a list of related CIs might be excessively long, which can slow performance when a CI form is rendered. You can configure these properties to control the amount of data that is displayed. To find a property, enter `sys_properties.list` in the left navigation filter and search for the property.

Properties related to performance

Property	Description
glide.ecmdb.find_relationship_issues	Hides or displays an icon in the CI relations formatter that links to open issues for the CI. This property defaults to true (displays the icon).
glide.ui.max_relation_levels	Specifies the maximum level for displaying CIs in flat view before reaching the maximum relations limit. The default value is 5.
glide.ui.max_relations	Specifies the maximum number of related CIs to display. When exceeded: <ul style="list-style-type: none">• A notification appears indicating that the limit has been reached, and that not all relations are displayed. The default value is 1000.• Flat layout reverts to the tree layout view.

- [Domain separation and the relations formatter and the CI relationship editor](#)

Domain separation is supported in the relations formatter and the CI relationship editor. Domain separation enables you to separate data, processes, and administrative tasks into logical groupings called domains. You can control several aspects of this separation, including which users can see and access data.

- [Create or edit a relationship filter](#)

Create a custom relationship filter to display CI relationships from selected tables in the CI relations formatter.

- [Exclude relationships from the relations formatter view](#)

Create a list of relationships that should not appear in the relations formatter view on CI forms.

- [Legacy CI relations formatter](#)

On instances that do not meet the internet browser requirements for the latest CI relations formatter, the default CI form includes the legacy CI relations formatter instead.

Related tasks

- [Add a suggested relationship](#)
- [Create a CI relation rollup](#)
- [Create or edit a relationship filter](#)

Related concepts

- [Relationship governance rules](#)
- [CI relationship editor](#)
- [Relation qualifier](#)
- [CI relationship security](#)

Related reference

- [Suggested class relationships](#)

Domain separation is supported in the relations formatter and the CI relationship editor. Domain separation enables you to separate data, processes, and administrative tasks into logical groupings called domains. You can control several aspects of this separation, including which users can see and access data.

Overview

How domain separation works in the relations formatter and relationship editor

- **Relations formatter**

The relations formatter is domain-separation supported. The relations formatter is used to display CMDB relationships in the UI in different views. Since the CI Relationship (cmdb_rel_ci) table is not domain separated, relationships are visible in the relations formatter only if both parent and child CIs (cmdb) are visible in the domain.

The CI Relationship Type (cmdb_rel_type) table is not domain separated. Therefore, in the relations formatter, all the relationship types are available to be selected as a filter.

By default domain separation is supported in the relations formatter.

- **Relationship editor**

The relationship editor is domain-separation supported. You can use the relationship editor to add new relationships or delete existing relationships for the current CI.

- The CI relationship editor displays a list of CIs to add or remove from relationships. Since they are domain separated, the CI list view in the relationship editor displays the CIs that are visible to the current domain.
- The CI relationship editor displays a list of relationships to add or remove. Since the CI Relationship [cmdb_rel_ci] table is not domain separated, the relationships list view displays all the relationships of the current CI.

The Suggested Relationship (cmdb_rel_type_suggest) table is not domain separated, which means that all the suggested relationship types in the relationship editor are visible for all domains.

By default domain separation is supported in the relationship editor.

Related concepts

- Domain separation and Configuration Management Database (CMDB)

Create a custom relationship filter to display CI relationships from selected tables in the CI relations formatter.

Before you begin

Role required: ecmdb_admin

About this task

The CI relations formatter displays related CIs for the base CI, and the relationships between the CIs. You can use relationship filters on the CI relations formatter to customize CI relationship views.

Procedure

1. Navigate to **All > Configuration > Relationships > Relationship Filters**.
2. Click **New** or select a filter to edit.
3. Enter or edit the relationship filter name.
4. Right-click the form header and click **Save**.
5. In the **Configuration Types** section, click **Edit**.
6. On the **Edit Members** form, select the tables of the CIs that you want to show with the filter and then move the tables to the **Configuration Types list**.
7. Click **Save**.

Result

On a CI form, in the relations formatter settings, you can select the newly defined relationship filter from the **Filter Relations by CMDB View** list.

In the legacy CI relations formatter, you can click **View** and select the newly defined relationship filter.

After you select a filter, the relations formatter displays only CIs from the tables specified in the filter or from descending tables.

Create a list of relationships that should not appear in the relations formatter view on CI forms.

Before you begin

Role required:

- To view the relationship type exclusion list — itil
- To create, update, or delete the relationship type exclusion list — itil_admin

Procedure

1. Navigate to **All > Configuration > Relationships > Relationship Type Exclusion List**.
2. In the CI Relation Filters list view, click **New**.
3. Fill out the CI Relation Filter form to specify the relationship that you want to exclude from view.
4. Click **Submit**.
Excluded relationships do not appear in Related Items on CI forms.

On instances that do not meet the internet browser requirements for the latest CI relations formatter, the default CI form includes the legacy CI relations formatter instead.

This element contains the list of related CIs and a toolbar with controls for viewing the relationships between the current CI and related CIs. For information about the latest CI relations formatter, see [CI relations formatter](#).

Related items field



Note: The legacy BSM map provides a more complete view of CI relationships.

Configure the controls in this formatter with two properties that restrict varying aspects of the view.

Flat layout

Click the flat layout icon (≡) to group the related Cls by relationship.

Flat layout view

The screenshot shows a list of related items under the heading "Related Items". The items are grouped by relationship:

- Used by - Business Services**
 - Bond Trading
 - Client Services
 - [Bond Trading] → IT Services
- In Rack - Racks**
 - NY-02-02
- Located in - Computer Rooms**
 - [NY2A] → NY Floor 2
- Runs - Web Servers**
 - apache linux den 200
- Located in Zone - Data Center Zones**
 - [NY-02-02] → NY2A

Tree layout

Click the tree layout icon (≡) to group the related Cls in a hierarchical tree.

Tree Layout View

The screenshot shows a hierarchical tree structure of related items:

- Linux100
 - NY-02-02
 - NY2A
 - NY Floor 2
 - New York

CI relationship editor

Use the relationship editor to view, create, modify, or delete CI relationships. Open the relationship editor from the CI Relations formatter.

When you use the relationship editor, the CI from which the editor was launched is designated as the base CI. You can then select one or more CIs as a second CI for the relationship. Depending on the selected relationship type, the base CI can become the parent CI or the child CI in the new relationship.

The relationship editor operates differently, depending on whether you select the **Use suggested relationship** check box.

- With suggested relationships, the relationship editor lists all available relationship types for the base CI. To define a new relationship, select a relationship type, and then select a second CI for the relationship.

Suggested relationships are highlighted for you. These relationships are displayed in blue with a prefix of [Suggested].

- Without suggested relationships, you define a new relationship by first selecting a second CI for the relationship and then selecting a parent or a child relationship type.

Note: The following relationship types are used only for Service Mapping endpoints, and you cannot use them as a relationship type between two CIs:

- Implement End Point To:Implement End Point From
- Use End Point To: Use End Point From
- Applicative Flow To:Applicative Flow From

Suggested relationships

If you select the **Use suggested relationship** check box in the editor, the **Suggested relationship** list appears. It displays all available CI, user and group relationship types for the base CI. Relationship types have a suffix of **(Parent)** or **(Child)** to note the relationship descriptor, and suggested relationship types are displayed in blue and have a "*" prefix.

When you select a relationship, you are also designating the base CI as being the parent or the child CI in the new relationship. For example, if you select the 'Feeds' relationship type, the base CI becomes the designated parent CI, and the second CI that you select becomes the child CI in this relationship.

Downstream relationships

If you do not select the **Use suggested relationship** check box in the editor, the **Downstream relationships** list appears. It displays all relationships in which the base CI is the parent CI. The child CI of the relationship is displayed in the **Child** column.

Upstream relationships

If you do not select the **Use suggested relationship** check box in the editor, the **Upstream relationships** list appears. It displays all relationships in which the base CI is the child CI. The parent CI in each relationship is displayed in the **Parent** column.

- [Create or edit a CI relationship](#)

Use the relationship editor to view, create, or modify CI relationships. You can open the relationship editor from the CI Relations formatter.

- [Delete a CI relationship](#)

Maintain the integrity of the CMDB by deleting any CI relationships that are no longer relevant or needed for a CI. Use the relationship editor to delete CI relationships.

- [Legacy CI relationship builder](#)

Used to define CI relationships manually, this page is a sophisticated version of the standard slushbucket. In the legacy CI relations formatter, click the CI relationship builder icon (+) to display the legacy Define Relationships page.

Related tasks

- [Add a suggested relationship](#)
- [Create a CI relation rollup](#)

- Add a suggested relationship
- Create or edit a CI relationship

Related concepts

- Relationship governance rules
- CI relations formatter
- Relation qualifier
- CI relationship security
- Legacy CI relationship builder

Related reference

- [Suggested class relationships](#)

Use the relationship editor to view, create, or modify CI relationships. You can open the relationship editor from the CI Relations formatter.

Before you begin

You must use supported browser versions in order to use the latest CI relationship editor. If you do not use a supported browser version, the instance provides the legacy CI relationship builder.

- Firefox version 20 and up
- Chrome version 25 and up
- Safari version 6 and up
- Internet Explorer version 9 and up

Role required:

- To create relationships: ITIL or asset
- To view relationships, depending on the state of the Table API ACL:
 - If inactive (default): ITIL or asset

- If active: ITIL or asset, and snc_platform_rest_api_access

For more information, see [REST API](#) and [Table API](#).

About this task

The relationship editor operates differently, depending on whether you check the **Use suggested relationship** option or not.

Procedure

- Launch the relationship editor:
 - Open a CI form.
 - Locate the **Related Items** section near the center of the form.
 - Click the plus (+) icon on the **Related items** section.
- To use suggested relationships, first select a relationship type, and then select one or more CIs to be the child CIs in the relationship:

Suggested relationship types

- Depends on (Parent)...
- Has (Child)...
- Member (Child)...
- Powered by (Child)...
- Receives data from (Parent)...
- Runs (Child)...

Filter

Class	is a	Computer	AND	OR	X
or Class	is a	Computer Peripheral			
Location	is anything		AND	OR	X
Operational status	is anything		AND	OR	X
Name	starts with	*Carol	AND	OR	X

Configuration Items

Name	Location	Description	Class	Updated	Maintenance schedule
*CAROL-IBM	Lenovo	322 West 52nd Street, New York, NY	Computer	2017-10-25 02:53:26	
*CAROL2-IBM	Lenovo	322 West 52nd Street, New York, NY	Computer	2017-10-25 02:53:12	
*CAROL3-GATEWAY	Gateway	322 West 52nd Street, New York, NY	Computer	2017-10-25 02:52:53	

Relationships

Type	Parent	Created
Depends on:Used by	abeherandez-snc-2017-10-10070721-buildtools1	2017-10-25 02:53:26
Depends on:Used by	abeherandez-snc-2017-10-10070721-buildtools1	2017-10-25 02:53:12
Depends on:Used by	abeherandez-snc-2017-10-10070721-buildtools1	2017-10-25 02:52:53

- Select **Use suggested relationship**.

- b. From the **Suggested relationship type** list, select a relationship type.
You can filter the list of suggested relationships by using the filter check boxes.

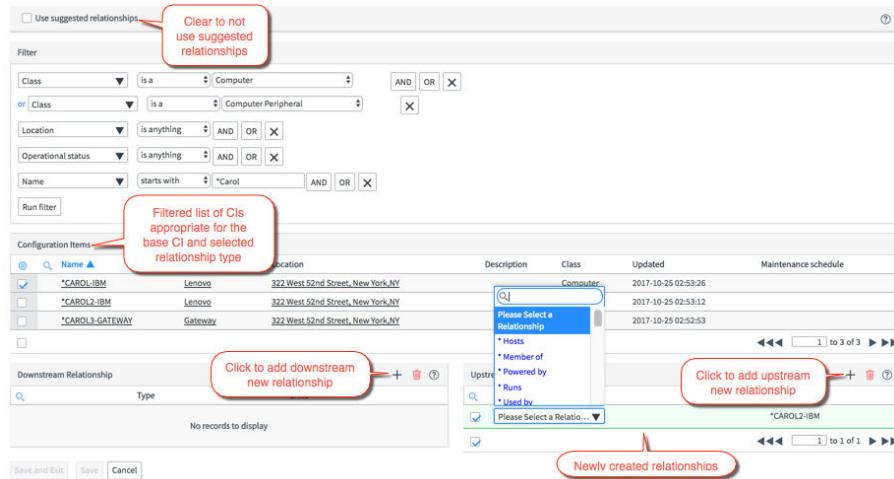
Filter option	Description
Hide CI relationship	Hides any relationships between the base CI and another CI (such as "Receives data from"). Default filter is stored in the ci_manage_relationships_filter_hint.cmdb_ci user preference.
Hide user relationship	Hides any relationships between the base CI and a user (such as "Logs reviewed by"). The default filter is stored in the ci_manage_relationships_filter_hint.sys_user preference.
Hide group relationship	Hides any relationships between the base CI and a group (such as "Backups done by"). Default filter is stored in the ci_manage_relationships_filter_hint.sys_user_group user preference.

The **Configurations Items** list displays all the CIs that are appropriate for the base CI and the selected relationship type. The **Relationships** list at the bottom of the editor, displays all existing relationships of the selected relationship type, in which the base CI is a parent CI or a child CI.

- c. From the **Configuration Items** list, select one or more CIs as a second CI for the relationship.
You can filter the list of **Configurations Items** by adding conditions in the **Filter** section and clicking **Run filter**.

If you selected a parent relationship type, these CIs becomes the child CI in the relationship, and if you selected a child relationship type, then the selected CIs become the parent CI in the relationship.

- d. In the **Relationships** section, click the plus icon (+) to add the new relationships.
Alternatively, you can drag the selected CIs to the **Relationships** list. Each new relationship will consist of the base CI, the selected relationship type, and a selected second CI.
3. To not use suggested relationships, first select one or more CIs to be the child CIs in the relationship, and then select the relationship type:



- a. Clear **Use suggested relationship**.
- b. In the **Configuration Items** list, select one or more CIs as a second CI for the relationship.
You can filter the list of **Configurations Items** by adding conditions in the **Filter** area and clicking **Run filter**. Depending on the relationship type that you will select, the selected CIs might become a parent or a child CI in the relationship.
- c. With at least one CI selected in the **Configuration Items** list, click the '+' sign in the **Downstream Relationships** section or the **Upstream Relationships** section to create the relationship.

- Add the relationship to **Downstream Relationships** to create a relationship in which the base CI is the parent CI and the selected CI is the child CI.
 - Add the relationship to **Upstream Relationships** to create a relationship in which the base CI is the child CI and the selected CI is the parent CI.
- d. For each newly created relationship in either the **Downstream Relationships** or the **Upstream Relationships** lists, click **Please select a relationship** and select a relationship type.
- The list of available relationship types in the **Downstream Relationships** list contains parent relationships only, in which the base CI is the parent CI.
 - The list of available relationship types in the **Upstream Relationships** list contains child relationships only, in which the base CI is the child CI.
- e. Click **Save** or **Save and Exit**.
Only after you enter all the information that is necessary for creating the relationship, these buttons light up indicating that there are pending updates that require saving.

Related concepts

- [CI relationships in the CMDB](#)
- [CI relationship security](#)

Related reference

- [Suggested class relationships](#)

Maintain the integrity of the CMDB by deleting any CI relationships that are no longer relevant or needed for a CI. Use the relationship editor to delete CI relationships.

Before you begin

Role required: ITIL or asset

Note: Deleting a relationship to a dependent CI can result in identification problems as the dependent CI will no longer have a relationship to the CI it depends on.

Procedure

1. Launch the relationship editor:
 - a. Open the CI form of the CI for which you want to delete a relationship.
 - b. Locate the **Related Items** section near the center of the form.
 - c. Click the plus (+) icon on the **Related items** section.
2. In the Relationship Editor, in the Relationships section, select the relationships that you want to delete for the CI.
3. Click the **Delete selected relationships** icon.
4. Click **Save** or **Save and Exit**.

Used to define CI relationships manually, this page is a sophisticated version of the standard slushbucket. In the legacy CI relations formatter, click the CI relationship builder icon () to display the legacy Define Relationships page.

For information about the latest CI relationship editor, see [CI relationship editor](#).

Select a CI relationship type

The top half of the legacy relationship editor contains a large option box that allows you to select which type of relationship you want to manipulate. Click the particular type of relationship you are interested in working with.

Filter the list of CI relationships

In the legacy relationship editor, the checkboxes along the right hand edge of the select box provide a quick way to filter down the list of available relationships.

By default, the system displays a list of all suggested relationships for the type of CI you selected. For example, if you selected a Database instance, a relationship of "Runs on" makes sense, but a relationship of "Provides HVAC for" does not. The default filter is stored in the user preferences

`ci_manage_relationships_filter_hint.cmdb_cici_manage_relationships_filter_hint.sys_user`, and `ci_manage_relationships_filter_hint.sys_user_group`.

- **Hide CI relationship** -- Hides any relationships between this CI and another CI (e.g. "Receives data from").
- **Hide user relationships** -- Hides any relationships between this CI and a user (e.g. "Logs reviewed by").
- **Hide group relationships** -- Hides any relationships between this CI and a group (e.g. "Backups done by").
- **Show all relationships** -- If you have the appropriate role (out of the box this is `itil_admin`) you will have an additional checkbox labeled "Show all relationships." If you click that checkbox, the system will let you choose any relationship defined in the system, regardless of where it is on the "suggested" list for this type of CI.

Select CI relationship targets

In the legacy relationship editor, users can link or unlink CIs for a relationship type.

As soon as you pick a relationship type, the system will fill in the two select boxes at the bottom of the screen with CIs that are appropriate for the relationship you suggested. The left hand select box will contain a list of CIs that might reasonably be linked via this relationship, while the right hand box contains a list of those CIs which are already linked.

1. Link or unlink items.

Link new items	Move that CI from the left hand box to the right hand box.
Unlink existing items	Move them from the right hand box to the left.

When you make either type of change, a message appears indicating that you have pending changes.

2. Apply or cancel your changes.

Click the Save button.

This will save your set of changes, and go back to the previous screen (either a CI or the BSM map depending on how you got here).

Click the Cancel button.

This causes you to exit without saving your changes.

Related concepts

- [CI relationships in the CMDB](#)

Relation qualifier

A relation qualifier, which is a CI of the Qualifier [cmdb_ci_qualifier] type, stores important information about the CI relationships.

In a relation qualifier, you can annotate arbitrary unique information about the relationship between two CIs. You can define multiple qualifiers for a single relationship, resulting in a qualifier chain. But, there can be only a single qualifier chain for a specific relationship type between two CIs.

For example, for a relationship between a parent CI and a child CI, you can add a relation qualifier to note that the relationship was discovered based on traffic (such as cmdb_ci_qualifier_trafficbased). This results in having two records in the CI Relationship [cmdb_rel_ci] table for the relationship.

- A record that links the parent CI and the new qualifier
- A record that links the new qualifier and the child CI

For this relationship, there is a parent CI and a child CI, and a relation qualifier of type cmdb_ci_qualifier_trafficbased.

For information about usage of relation qualifiers in the identification process, see [Identification rules](#).

Related tasks

- [Add a suggested relationship](#)
- [Create a CI relation rollup](#)

Related concepts

- [Relationship governance rules](#)
- [CI relations formatter](#)
- [CI relationship editor](#)
- [CI relationship security](#)

Related reference

- [Suggested class relationships](#)

CI relationship security

When applying security to CI relationships, it is important to apply the access controls both to the CI Relationship (cmdb_rel_ci) table and to create an operation editCIRelations to the * table as well.

If the current instance has defined security for editCIRelations, it will be applied to edit_ci_relations automatically in the process of upgrading, and the out-of-date security will be removed.

Related tasks

- [Add a suggested relationship](#)
- [Create a CI relation rollup](#)
- [Create a CI relation rollup](#)

Related concepts

- [Relationship governance rules](#)
- [CI relations formatter](#)
- [CI relationship editor](#)
- [Relation qualifier](#)
- [CI relationships in the CMDB](#)

Related reference

- [Suggested class relationships](#)

Create a CI relation rollup

A CI relation rollup allows you to sum, count, max, min, or mean a relationship type. You can create CI relation rollups.

Before you begin

Role required: ecmdb_admin

About this task

CI relation rollup can be useful for tracking and for receiving notifications. For example:

- In a sum roll up, add up fields from multiple CIs and display the result on another CI to which they are related. So, if you have four configuration items in a rack that are all consuming power, create a CI relation rollup to add all the power usage together and display the result in one field on the rack CI form.
- If a certain level of power consumption in a rack is exceeded, send a notification.
- With a rack that has 10 slots, send a notification when 9 slots are filled.

CI relation rollups use the cmdb synch event business rule on the [cmdb_ci] table. Although this business rule is active by default, you must modify the rule slightly before it will run.

Procedure

1. Navigate to **All > Configuration > Relationships > CI Relation Rollups**.
2. Click **New**.
3. Complete the form.

CI Relationship Rollup fields

Field	Description
CI Relationship Type	Select a relationship type from the list to use with the rollout. For example, Members::Member of contains the parent descriptor Members and the child descriptor Member of .
Type	Select the type of rollout from the drop-down list: COUNT, MAX, MEAN, MIN, or SUM.
Parent field	The target field on which the operation will be done.
Child field	The input to the equation type. The Parent field is affected by the selections in the child field.
Rollup class	The classes that can use the relationship. For example, you can specify that the relationship only applies to racks.

4. To run the cmdb synch event business rule, navigate to **Business Rules**.
5. Use the search box to find the [cmdb synch event] table.

6. Click the cmdb synch event business rule to go to the **Business Rule** page.

7. Select the **Update**, **Delete**, and **Query** check boxes.

Additionally, if you wish CI relation rollups to recalculate when there is a change to a relationship, use a similar procedure to select the **Active** check box on the cmdb_rel_ci synch event business rule.

Related tasks

- [Add a suggested relationship](#)

Related concepts

- [Relationship governance rules](#)
- [CI relations formatter](#)
- [CI relationship editor](#)
- [Relation qualifier](#)
- [CI relationship security](#)
- [CI relationships in the CMDB](#)

Related reference

- [Suggested class relationships](#)

CMDB schema model

The Configuration Management Database (CMDB) schema model is a series of connected tables that contain all the assets and business services controlled by a company and its configurations.

Related ServiceNow® Store apps and reference information:

- [CMDB tables descriptions](#): Descriptions of key CMDB tables in the base system.
- [CMDB CI Class Models](#): A ServiceNow Store app that adds class models that extend the base CMDB class hierarchy. This includes

class descriptions, identification rules, identifier entries, and dependent relationships if applicable. You can then use the added classes as any other CMDB base class.

- **Populating the CMDB:** Information about the various options for populating the CMDB.
- **Discovery patterns:** A ServiceNow Store app that provides a library of Discovery patterns for discovering specific devices and applications in the industry.
- **Service Graph Connectors:** ServiceNow Store apps that provide pre-defined integrations for importing and integrating common third-party data into CMDB classes. Also includes the [IntegrationHub ETL](#) wizard for creating new ETL transform maps.

CMDB tables contain information about computers and devices on the network, software contracts and licenses, business services, and so on. The IT desk can use the CMDB to better understand their network users' equipment, and the relationships between them. The CMDB can also be referenced by other processes within the system.

Applications such as Asset Management and Contract Management, operate in conjunction with the CMDB. Asset Management and Software Asset Management link to CMDB all assets, hardware, software, assets in stock, as well as records for manufacturers and vendors. The Contract Management application contains information about contracts, including leases, service contracts, purchase orders, warranties, and software licenses. The Configuration Management Database (CMDB) application has a focus on operation.

For more background information about the CMDB, see the ServiceNow Community post) at [CMDB 101- What is a configuration management database and why do you need one?](#).

Key CMDB tables

Key tables in the configuration management database (CMDB):

- The Base Configuration Item [cmdb] table, which is the core CMDB table for non IT CIs (descending classes are non IT CIs).

- The core Configuration Item [cmdb_ci] table, which stores the basic attributes of all the CIs. The admin, itil, or asset user role is required to access this table (descending classes are IT CIs).
- The CI Relationship [cmdb_rel_ci] table, which defines all relationships between CIs.

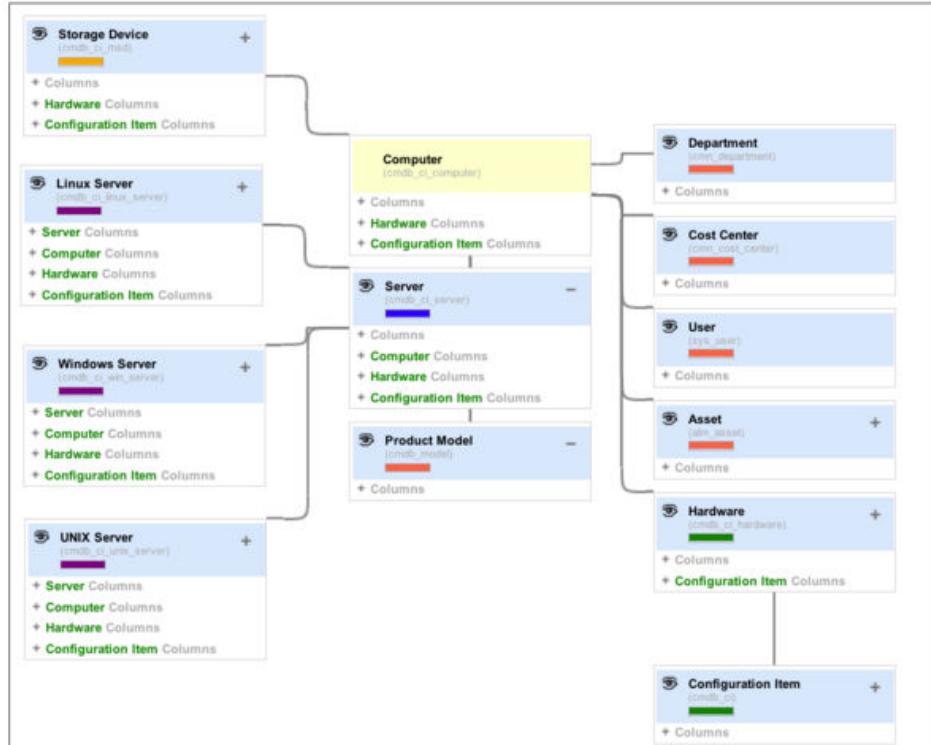
The Configuration Item table is extended to other tables, such as Database [cmdb_ci_database] and Computer [cmdb_ci_computer]. The Computer table is extended to the Server [cmdb_ci_server] table, which is extended to the UNIX Server [cmdb_ci_unix_server] table, and so on.

Note: The Base Configuration Item [cmdb] table uses the table per partition extension model, which has different behaviors for replicating and deriving information than other extended tables. See [Table extension and classes](#).

You can use the schema map to view more details of tables and their relationships:

1. Navigate to **System Definition > Tables & Columns**.
2. Select a table and click **Schema Map**.

Schema Map



Note: CIs not extended from the Configuration Item [cmdb_ci] table, are not displayed in Dependency Views maps and in CI relation formatters.

CI attributes

Attributes apply to all the CIs in a classification. To change attribute values for a CI, edit the appropriate CI. To add a unique attribute to a class, extend the class table and create a new classification for that CI.

The position of a CI in a classification hierarchy is determined by the attributes it shares with the CIs below it. Each time a CI has a single different attribute from its parent, the classification hierarchy branches.

For example, servers have different attributes from computers, which include workstations and laptops. Linux servers and UNIX servers have different attributes from the parent server classification and from each other, so they occupy separate branches in the hierarchy.

CMDB tables descriptions

List of tables in the CMDB in a base system with its name, label, and a description of the type of information that is stored in the table.

You can extend tables in a base system by installing a CMDB CI Class Models store app which adds class models that support specific technologies. These extensions include class definitions, identification rules, identifier entries, and dependent relationships if applicable. For more information, see [CMDB CI Class Models store app](#).

Table name	Table label	Table description
cmdb_ci	Configuration Item	Base configuration item table.
cmdb_ci_acc	Accessory	Accessories for phones, computers, and so on.
cmdb_ci_ad_controller	Active Directory Domain Controller	Microsoft Active Directory domain controller.
cmdb_ci_ad_domain	AD Domain	Microsoft Active Directory domain.
cmdb_ci_aix_server	AIX Server	Server running the AIX operating system.
cmdb_ci_alias	Alias	Pseudonym for data locations, virtual email addresses, pointers, and so on.
cmdb_ci_apache_web_server	Apache Web Server	Server hosting Apache web server software.
cmdb_ci_appl	Application	Application, which is a collection of files and

Table name	Table label	Table description
		data that deliver a service and manage business processes.
cmdb_ci_appl_now_app	ServiceNow Application	<p>CIs that Event Management generates for various components such as Impact calculator, which are used to bind alerts that are later shown in maps. Used internally by self-health when monitoring internal health checks for key components such as connector instance status and MID Server status.</p> <p>Parent class for all ServiceNow applications.</p>
cmdb_ci_application_cluster	Application Cluster	Logical group of servers with clustering software installed on each of the servers in the group so that the group acts like a single system.
cmdb_ci_application_server_resource	Application Server Resource	Parent class for application servers such as Coldfusion application server.

Table name	Table label	Table description
cmdb_ci_app_server	Application Server	A base table for logical CIs, which indicate the primary function of a physical or virtual server such as a Tomcat server or a WebSphere server.
cmdb_ci_app_server_composer	Composer	Server hosting IBM WebSphere Multichannel Bank Transformation Toolkit.
cmdb_ci_app_server_datapower	Data Power	Server hosting IBM DataPower Gateway Secure software.
cmdb_ci_app_server_domino	Domino	Server hosting IBM Domino software.
cmdb_ci_app_server_hp_ucmdb	HP uCMDB	Server hosting HP uCMDB software.
cmdb_ci_app_server_java	JavaServer	Server hosting Java application.
cmdb_ci_app_server_jb_module	delivery Controller	Server hosting inner module of JBoss application (deployed application).
cmdb_ci_app_server_jboss	JBoss	Server hosting JBoss Application Server (JBoss AS), which is a cross-platform Java application server, open-source developed by JBoss software company.

Table name	Table label	Table description
cmdb_ci_app_server_jrun	Jrun	Server hosting JRun application.
cmdb_ci_app_server_jrun_war	Jrun WAR	Server hosting the inner module of JRun application (deployed application).
cmdb_ci_app_server_ora_ess	Oracle Essbase Server	Server hosting Oracle Essbase software.
cmdb_ci_app_server_ora_ias	Oracle iAS	Server hosting Oracle Internet Application Server.
cmdb_ci_app_server_ora_ias_m	Oracle iAS Web module	Server hosting the inner module of Oracle iAS application (deployed application).
cmdb_ci_app_server_remedy	Remedy HSServer	Server hosting Remedy HSServer application.
cmdb_ci_app_server_tomcat	Tomcat	Server hosting Apache Tomcat software.
cmdb_ci_app_server_tomcat_war	Tomcat WAR	Server hosting inner module of Apache Tomcat application (deployed application).
cmdb_ci_app_server_vendavo	Vendavo Application Server	Server hosting Vendavo Application Server software.
cmdb_ci_app_server_weblogic	BEA Weblogic	Server hosting Oracle WebLogic Server.

Table name	Table label	Table description
cmdb_ci_app_server_webseal	Webseal	Server hosting IBM Tivoli Access Manager solution.
cmdb_ci_app_server_websphere	IBM Websphere	Server hosting IBM WebSphere software.
cmdb_ci_app_server_wl_module	WeblogicModule	Server hosting inner module of Tomcat.
cmdb_ci_app_server_ws_ear	Websphere EAR	Server hosting inner module of IBM WebSphere software.
cmdb_ci_app_server_ws_odr	Websphere ODR LB	Server hosting WebSphere ODR LB application.
cmdb_ci_appl_active_directory	Active Directory Service	Inner software module of AD Domain application.
cmdb_ci_appl_biztalk	BizTalk	Microsoft BizTalk Server software.
cmdb_ci_appl_biztalk_orch	BizTalk Orchestration	Inner module of Microsoft Biztalk Server software.
cmdb_ci_appl_ca	CA Enterprise Communicator	CA Enterprise Communicator software.
cmdb_ci_appl_ca_dir_server	CA eTrust Directory Server	CA eTrust Directory Server software.
cmdb_ci_appl_ca_ent_man	CA Introscope Enterprise Manager	CA introscope Enterprise Manager software.

Table name	Table label	Table description
cmdb_ci_appl_ca_id_man	CA Identity Manager Provisioning Server	CA Identity Manager Server software.
cmdb_ci_appl_cisco_call_ma n	Cisco CallManager	Cisco CallManager (Cisco Unified Communications Manager) software.
cmdb_ci_appl_cisco_fibre	Cisco Fibre InterConnect	Cisco Fibre InterConnect software.
cmdb_ci_appl_citrix_app	Citrix Application Icon	Inner module of Citrix software.
cmdb_ci_appl_citrix_collector	Citrix Collector	Citrix Collector software.
cmdb_ci_appl_citrix_xenapp	Citrix XenAPP or Presentation Server	Citrix XenApp software.
cmdb_ci_appl_connectit	Connect-It Service	Connect-It software.
cmdb_ci_appl_controlm	Control-M	Control-M software.
cmdb_ci_appl_delivery_contr oler	Delivery Controller	Application delivery controller software.
cmdb_ci_appl_doc_brava_pr oc	Documentum Brava Job Processor	Brava (EMC Documentum) job processor software.
cmdb_ci_appl_doc_brava_se rver	Documentum Brava License Server	Brava (EMC Documentum) License Server software.

Table name	Table label	Table description
cmdb_ci_appl_doc_docbase	Documentum DocBase	Documentum Docbase software.
cmdb_ci_appl_doc_docbroker	Documentum Broker	Documentum Docbase broker software.
cmdb_ci_appl_dot_net	.NET Application	Microsoft .NET application software.
cmdb_ci_appl_fastsearch	Fast Search	Microsoft FAST Search software (for the SharePoint collaboration platform).
cmdb_ci_appl_generic	Generic Application	Generic application, which is identified by the system when there is an endpoint with an open port in listen mode and there is no pattern for it.
cmdb_ci_appl_glassfish	GlassFish	Oracle GlassFish Server software.
cmdb_ci_appl_glassfish_war	GlassFish WAR	Inner module of GlassFish application (deployed application).
cmdb_ci_appl_groundwork	Groundwork	Groundwork (open source) monitoring software.
cmdb_ci_appl_hp_index	HP SM Index Server	HP Service Manager Index Server software.

Table name	Table label	Table description
cmdb_ci_appl_hp_operations	HP Operations Manager	HP Operations Manager software.
cmdb_ci_appl_hp_qc	HP Quality Center	HP Quality Center software.
cmdb_ci_appl_hp_service	HP Service Manager	HP Service Manager software.
cmdb_ci_appl_hp_sm_kb	HP SM KnowledgeBase	HP Service Manager KnowledgeBase software.
cmdb_ci_appl_ibm_cics	IBM CICS	IBM CICS Transaction Server software.
cmdb_ci_appl_ibm_ctg	IBM CTG	IBM CICS Transaction Gateway software.
cmdb_ci_appl_ibm_wmb	IBM WebSphere Message Broker	IBM WebSphere Message Broker software.
cmdb_ci_appl_ibm_wmb_listener	IBM WMB Http Listener	IBM WebSphere HTTP Listener software.
cmdb_ci_appl_ibm_wmq	IBM WebSphere MQ	IBM Websphere MQ software.
cmdb_ci_appl_ibm_wmq_queue	IBM WebSphere MQ Queue	Inner module of IBM WebSphere MQ software.
cmdb_ci_appl_itam	ITAM Asset Center	HP Asset Center software.
cmdb_ci_appl_mongo_config_serv	Mongo Config Server	Mongo Configuration Server software.

Table name	Table label	Table description
cmdb_ci_appl_mongos	Mongos Server	MongoDB server software.
cmdb_ci_appl_ms_dynamic_crm	Dynamic CRM Component	Microsoft Dynamic CRM software.
cmdb_ci_appl msmq	MSMQ	Microsoft Message Queuing (MSMQ) software.
cmdb_ci_appl_ora_conc	Oracle Concurrent Server	Oracle Concurrent Server software.
cmdb_ci_appl_ora_disc	Oracle Discoverer Engine	Oracle Discoverer software.
cmdb_ci_appl_ora_disc_ui	Oracle Discoverer UI	Oracle Discoverer UI module software.
cmdb_ci_appl_ora_ebs	Oracle ESB	Oracle Enterprise Service Bus software.
cmdb_ci_appl_ora_forms	Oracle Forms Engine	Oracle Forms software.
cmdb_ci_appl_ora_forms_ui	Oracle Forms UI	Oracle Forms UI software.
cmdb_ci_appl_ora_fs	Oracle Fulfillment Server	Oracle Fulfillment Server software.
cmdb_ci_appl_ora_http	Oracle HTTP Server	Oracle HTTP Server software (web tier of Oracle Fusion middleware).

Table name	Table label	Table description
cmdb_ci_appl_ora_jms_queue	Oracle Weblogic JMS Queue	Oracle WebLogic JMS software.
cmdb_ci_appl_ora_metric_client	Oracle Metric Client	Oracle Metric client software.
cmdb_ci_appl_ora_metric_svr	Oracle Metric Server	Oracle Metric server software.
cmdb_ci_appl_ora_notif_svr	Oracle Notification Server	Oracle Notification Server (ONS) software.
cmdb_ci_appl_ora_oacore	Oracle OACORE Server	Oracle OACORE server software.
cmdb_ci_appl_ora_oafm	Oracle OAFM Server	Oracle OAFM server software.
cmdb_ci_appl_ora_pm	Oracle Process Manager	Oracle BPEL Process Manager server software.
cmdb_ci_appl_ora_queue	Advanced Queue Queue	Oracle Advanced Queuing software.
cmdb_ci_appl_ora_report	Oracle Report Server	The Oracle Report Server software.
cmdb_ci_appl_ora_tns	Oracle App TNS Service	Oracle Application Express (TNS) listener software.
cmdb_ci_appl_ora_tnslsnr	Oracle TNS Listener Engine	Oracle Application Express software.

Table name	Table label	Table description
cmdb_ci_appl_peoplesoft	PeopleSoft Application Server	PeopleSoft Application Server software.
cmdb_ci_appl_rabbitmq	RabbitMQ	RabbitMQ (open source) software.
cmdb_ci_appl_rabbitmq_cluster	RabbitMQ Cluster	RabbitMQ Cluster (open source) software.
cmdb_ci_appl_sap_asc	SAP ASCS Application	SAP ASCS software.
cmdb_ci_appl_sap_bo	SAP Business Objects CMS Server	SAP Business Objects CMS server software.
cmdb_ci_appl_sap_bo_sched	SAP BO BOXIScheduleRouter	SAP BO BOXIScheduleRouter software.
cmdb_ci_appl_sap_bus_obj	SAP Business Objects	SAP business Object application.
cmdb_ci_appl_sap_ci	SAP CI Application	The SAP Central Instance software.
cmdb_ci_appl_sap_di	SAP DI Application	Oracle Development Infrastructure (DI) software.
cmdb_ci_appl_sap_ers	SAP ERS Application	Oracle Evaluated Receipt Settlement (ERS) software.
cmdb_ci_appl_sap_hana_db	SAP Hana Db	SAP HANA software.
cmdb_ci_appl_sap_jc	SAP JC Application	SAP JC (java application) software.

Table name	Table label	Table description
cmdb_ci_appl_sap_scs	SAP SCS Application	SAP SCS (central services) software.
cmdb_ci_appl_sendmail	Sendmail	Sendmail (open source) software.
cmdb_ci_appl_sharepoint	SharePoint	Microsoft SharePoint software.
cmdb_ci_appl_sp_service	SharePoint Service	Microsoft Windows Sharepoint Services (WSS) software.
cmdb_ci_appl_tibco_hawk	Tibco Hawk	TIBCO Hawk software.
cmdb_ci_appl_tibco_matrix	ActiveMatrix Business Works	TIBCO ActiveMatrix BusinessWorks software.
cmdb_ci_appl_tibco_matrix_proc	ActiveMatrix Business Works Process	TIBCO ActiveMatrix BusinessWorks Process software.
cmdb_ci_appl_tibco_message	Tibco Enterprise Message Service	TIBCO Enterprise Message Service software.
cmdb_ci_appl_tibco_queue	EMS Queue	Tibco EMS (Enterprise Message Service) Queues software.
cmdb_ci_appl_tuxedo	Tuxedo	Tuxedo software (middleware transactions for Unix, Extended for Distributed Operations).
cmdb_ci_appl_tuxedo_portal	Tuxedo Portal	Tuxedo portal software.

Table name	Table label	Table description
cmdb_ci_appl_vign_content_svr	Vignette Content Management Server	Vignette (Open Text Corp) Content Management Server software.
cmdb_ci_appl_vignette_search	Vignette Search Starter	Vignette (Open Text Corp) Search Server software.
cmdb_ci_appl_vignette_server	Vignette Server	Vignette (Open Text Corp) Server software.
cmdb_ci_appl_weblogic_jms	Weblogic JMS Server	WebLogic JMS software.
cmdb_ci_appl_weblogic_lb	Weblogic LB	WebLogic Server load balancer software.
cmdb_ci_appl_weblogicmodule	Weblogic Module Server	WebLogic Server software.
cmdb_ci_appl_websphere_portal	Websphere Portal	WebSphere Portal software.
cmdb_ci_appl_wmb	WMB Flow	WebSphere Message Broker software.
cmdb_ci_application_cluster	Application Cluster	Logical cluster of application-tier servers.
cmdb_ci_application_software	Application Software	<p>Computer program designed to perform a group of coordinated functions, tasks, or activities for the benefit of the user.</p> <p>An extension of the Software table,</p>

Table name	Table label	Table description
		providing installed software information (not a running process).
cmdb_ci_ats_power_eq	Automatic Transfer Switch	Electrical power switch that switches a load between two sources.
cmdb_ci_availability_set	Availability Set	Logical grouping of virtual machines running on Microsoft Azure platform.
cmdb_ci_batch_job	Batch Job	A computer program or set of programs processed in batch mode.
cmdb_ci_aws_datacenter	AWS datacenter	Logical representation of an Amazon Web Services datacenter.
cmdb_ci_azure_datacenter	Azure datacenter	Logical representation of a Microsoft Azure datacenter.
cmdb_ci_business_app	Business Application	All business applications.
cmdb_ci_business_process	Business Process	A process that is owned and carried out by the business and contributes to the delivery of a product or business service to a business customer.
cmdb_ci_chassis_server	Server Chassis	A metal structure that is used to house or

Table name	Table label	Table description
		physically assemble servers in various different form factors.
cmdb_ci_cim_profile	CIM Profiles	CIM Profiles (UML).
cmdb_ci_cim_server	CIM Server	Server hosting CIM profiles.
cmdb_ci_circuit	Circuit	Electrical circuits information.
cmdb_ci_cloud_database	Cloud Database	Database which runs on a cloud computing platform.
cmdb_ci_cloud_ip_address	Cloud IP Address	Web server which runs on a cloud computing platform.
cmdb_ci_cluster	Cluster	Logical group of computing resources bound together by software in order to function as one logical computing resource.
cmdb_ci_cluster_node	Cluster Node	Single computing resource which is logically/operationally bound into a cluster.
cmdb_ci_cluster_resource	Cluster Resource	System object that is a set or grouping of cluster resources that are used to manage events that occur in a clustered environment.

Table name	Table label	Table description
cmdb_ci_cluster_vip	Cluster Virtual IP	Cluster VIP information.
		<p>Used to calculate the approximate monthly cost of running a stack built on Virtual Servers in cloud environment:</p> <ul style="list-style-type: none"> • Cloud Price Base: Common price base to store pricing info of all the resources of all the supported clouds.
cmdb_ci_cmp_price_product_base	Cloud Product Price Base	<ul style="list-style-type: none"> • Cloud Price Product Base: Base table to store the common attributes of all the pricing resources. • VM Instance Price: Specific table which extends the Cloud Price Product base to store the VM specific product info of all the supported clouds. <p>Parent class for cloud product prices such as VM instance price.</p>
cmdb_ci_cmp_resource	Cloud Resource	Generic cloud resources.
cmdb_ci_comm	Communication Device	Communication devices information. A

Table name	Table label	Table description
		choice list containing devices such as cellphone, phone, conference phone, and Wi-Fi.
cmdb_ci_computer	Computer	An extension of the Hardware table, capturing computer properties.
cmdb_ci_computer_room	Computer Room	Logical representation of a computer room.
cmdb_ci_config_automation_server	Management Server	Dev Ops tools such as Chef and Puppet, that are used to manage server configurations. Parent class for application management servers such as Puppet Master.
cmdb_ci_config_file	Configuration file	Configuration files which establish the parameters and initial settings for some computer programs.
cmdb_ci_crac	Computer Room AC	Air conditioning units used to cool data centers.
cmdb_ci_csu_dsu_network	CSU/DSU	Digital-interface device used to connect networking

Table name	Table label	Table description
		equipment to a digital circuit.
cmdb_ci_database	Database	Organized collection of data such as the set of files where data is stored, the reason for a database, and the metadata about the data.
cmdb_ci_datacenter	Data Center	Facility used to house computer systems and associated components, such as telecommunications and storage systems. It generally includes redundant or backup power supplies, redundant data communications connections, environmental controls (such as air conditioning and fire suppression), and various security devices.
cmdb_ci_datapower_server	Data Power Hosting Server	Server running IBM DataPower Gateway software.
cmdb_ci_datastore	Datastore	Datastores are like file systems, abstracting the physical storage and providing a model for storing files. The child class represents the

Table name	Table label	Table description
		VMware specific Datastore. Parent class for VMWare datastore object types such as vCenter Datastores.
cmdb_ci_db_catalog	Database Catalog	Metadata which defines database objects such as base tables, views (virtual tables), synonyms, value ranges, indexes, users, and user groups, for a specific database instance.
cmdb_ci_db_db2_catalog	DB2 Catalog	Database catalog for DB2 database.
cmdb_ci_db_db2_instance	DB2 Instance	Instance of a DB2 database.
cmdb_ci_db_hbase_instance	HBase Instance	Instance of an HBase database.
cmdb_ci_db_instance	Database Instance	Software and memory used to manipulate data in a database.
cmdb_ci_db_mongodb_instance	MongoDB Instance	Instance of a MongoDB database.
cmdb_ci_db_mssql_analysis	SQL Server Analysis Services	Microsoft SQL Server Analysis Services software.
cmdb_ci_db_mssql_catalog	MSFT SQL Catalog	Database catalog for a specific instance

Table name	Table label	Table description
		of a Microsoft SQL database.
cmdb_ci_db_mssql_int_job	SQL Server Integration Services Job	Scheduled job to run a SQL Server Integration Service package.
cmdb_ci_db_mssql_integration	SQL Server Integration Services	MSQL Server Integration Services software.
cmdb_ci_db_mssql_reporting	SQL Server Reporting Services	SQL Server software used for server-based reporting generation.
cmdb_ci_db_mssql_server	MS SQL Server	Microsoft SQL Server.
cmdb_ci_db_mysql_catalog	MySQL Catalog	Database catalog for a specific instance of a MySQL database.
cmdb_ci_db_mysql_clustermg node	MySQLClusterGMNode	MySQL primary administrative interface to a running cluster.
cmdb_ci_db_mysql_clusterno de	MySQLClusterDataNode	Summary table used in the [ndbd] or [ndbd default] sections of a config.ini file for configuring MySQL Cluster data nodes.
cmdb_ci_db_mysql_instance	MySQL Instance	Instance of a MySQL database.
cmdb_ci_db_ora_catalog	Oracle Catalog	Database catalog for a specific instance of an Oracle database.

Table name	Table label	Table description
cmdb_ci_db_ora_instance	Oracle Instance	Instance of an Oracle database.
cmdb_ci_db_ora_listener	Oracle Database Listener	Process that runs on an Oracle Database Server.
cmdb_ci_db_postgresql_instance	PostgreSQL Instance	Instance of a PostgreSQL database.
cmdb_ci_db_syb_catalog	Sybase Catalog	Database catalog for a specific instance of a Sybase database.
cmdb_ci_db_syb_instance	Sybase Instance	Instance of a Sybase database.
cmdb_ci_desktop_software	Desktop Software	Software used on desktops and laptops.
cmdb_ci_dir_policy_server	Policy Server	Policy server, which provides a security component of a policy-based network that provides authorization services and facilitates tracking and control of files.
cmdb_ci_dir_site_minder_server	Site Minder	Server running SiteMinder software.
cmdb_ci_directory_ad_forest	AD Forest	Active Directory forest.
cmdb_ci_directory_ha	HA Proxy	HAProxy software.
cmdb_ci_directory_iifp	IIFP	Identity Identification Feature Pack (Active Directory) software.

Table name	Table label	Table description
cmdb_ci_directory_ldap	LDAP DB	LDAP (Lightweight Directory Access Protocol) database software.
cmdb_ci_directory_server	Directory Server	Server running LDAP software.
cmdb_ci_disk	Disk	General category of data storage mechanisms.
cmdb_ci_disk_partition	Disk Partition	Sections of a disk separated so that information in each section can be managed separately.
cmdb_ci_display_hardware	Display Hardware	Hardware used to display information in visual form.
cmdb_ci_dns_alias	DNS Alias	Synonym for the host used to resolve DNS addresses.
cmdb_ci_dns_name	DNS Name	Primary DNS names.
cmdb_ci_docker	Docker Container	Docker containers (a runtime instance of a docker image).
cmdb_ci_docker_engine	Docker Engine	Docker software for running and managing Docker containers.
cmdb_ci_docker_image	Docker Image	Docker images. Ordered collection of root filesystem changes and

Table name	Table label	Table description
		the corresponding execution parameters for use within a container runtime.
cmdb_ci_docker_image_tag	Docker Image Tag	Docker tag, which is a label applied to a Docker image in a repository.
cmdb_ci_docker_local_image	Docker Local Image	Locally managed Docker image.
cmdb_ci_drs_vm_config	DRS VM Config	Distributed Resource Scheduler (DRS) behavior for the VMs in the vCenter that override the cluster behavior.
cmdb_ci_ec2_instance	EC2 Virtual Machine Instance	Virtual machine running in the Amazon Elastic Compute Cloud (EC2) platform.
cmdb_ci_email_server	Email Server	Server running email software.
cmdb_ci_email_server_jes	JES	Server running JES software (multi-featured hybrid MTA/MDA server).
cmdb_ci_endpoint	Endpoint	Endpoint, which represents the entry point to a service, a process, or a queue or topic destination in service-oriented architecture.

Table name	Table label	Table description
cmdb_ci_environment	Environment	Logical grouping of hardware and software used to develop, test, and deliver computing services. For example: development, test, quality assurance, and production.
cmdb_ci_esx_resource_pool	ESX Resource Pool	VMware set of physical resources.
cmdb_ci_esx_server	ESX Server	Physical ESX server running the VMware ESXi operating system.
cmdb_ci_exchange_backend	ExchangeBackEndServer	Server running Exchange software.
cmdb_ci_exchange_cas	Exchange Client Access Server	Server running Exchange software providing client access services.
cmdb_ci_exchange_edge_transport_server	Exchange Edge Transport Server	Server running Exchange Edge Transport software.
cmdb_ci_exchange_frontend	ExchangeFrontEndServer	Server running Exchange software.
cmdb_ci_exchange_hub	ExchangeHub	Server running Exchange Hub software.
cmdb_ci_exchange_hub_transport_server	Exchange Hub Transport Server	Server running Exchange Hub software providing transport services.

Table name	Table label	Table description
cmdb_ci_exchange_mailbox	Exchange MailBox	Exchange email account.
cmdb_ci_exchange_mailbox_server	Exchange Mailbox Server	Server running Exchange software providing client access services.
cmdb_ci_exchange_service_component	Exchange Service Component	Exchange Service Component software.
cmdb_ci_facility_hardware	Facility Hardware	Base class for hardware used to facilities services such as electric, water, sewer, air, and security.
cmdb_ci_fc_disk	Fibre Channel Disk	Base table for fibre channel disk.
cmdb_ci_fc_export	Fibre Channel Export	Storage volume exported by a storage server via Fibre Channel protocol.
cmdb_ci_fc_port	Fibre Channel Port	Fibre Channel port on a storage server, FC switch, or on a host's HBA.
cmdb_ci_fddi_network	FDDI Cards	Fiber Distributed Data Interface cards.
cmdb_ci_file_system	File System	File system information for a server, capturing details such as mount point, capacity, and type of file system.

Table name	Table label	Table description
cmdb_ci_file_system_nfs	NFS File system	Extension of File System, which provides NFS file system information.
cmdb_ci_file_system_smb	SMB File system	Extension of File System, which provides SMB file system information.
cmdb_ci_firewall_network	Firewall Hardware	Firewall hardware.
cmdb_ci_ftp_server	FTP Server	Server providing FTP services.
cmdb_ci_fuel_tank	Fuel Tank	Fuel tank.
cmdb_ci_generator_power_eq	Power Generator	Power generator.
cmdb_ci_group	Group	Logical group of Cls.
cmdb_ci_hardware	Hardware	Base class for hardware.
cmdb_ci_host_cluster	Host Cluster	<p>Cloud agnostic way of representing a group of hosts as a cluster. The child class represents VMware vCenter Cluster specific details.</p> <p>Parent class for VMWare host cluster object types such as vCenter Clusters.</p>

Table name	Table label	Table description
cmdb_ci_hpx_server	HPUX Server	Server running HPUX software.
cmdb_ci_hub_network	Hub Hardware	Physical network hub.
cmdb_ci_hvac	HVAC Equipment	Heating, ventilation, and air conditioning equipment.
cmdb_ci_hyper_v_cluster	Hyper-V Cluster	Cluster of the Hyper-V servers.
cmdb_ci_hyper_v_instance	Hyper-V Virtual Machine Instance	Hyper-V virtual machine instance. This table extends the generic Virtual Machine Instance [cmdb_ci_vm_instance] table.
cmdb_ci_hyper_v_network	Hyper-V Virtual Network	Hyper-V virtual network.
cmdb_ci_hyper_v_object	Hyper-V Object	Base class for all Hyper-V objects.
cmdb_ci_hyper_v_resource_pool	Hyper-V Resource Pool	Hyper-V resource pool.
cmdb_ci_hyper_v_rpool_component	Hyper-V Resource Pool Component	Resource pool component belonging to resource pool.
cmdb_ci_hyper_v_server	Hyper-V Server	Server running Hyper-V software.
cmdb_ci_ids_network	Intrusion Detection System	Security intrusion detection systems.

Table name	Table label	Table description
cmdb_ci_iisdirectory	IIS Virtual Directory	Virtual Directory in IIS Manager.
cmdb_ci_imaging_hardware	Imaging Hardware	Hardware used to create electronic/physical images.
cmdb_ci_inetinfo	Inetinfo service	Inetinfo service of IIS application.
cmdb_ci_inf_software	Infrastructure Software	Base class for enterprise software or programs specifically designed to help business organizations perform basic tasks such as workforce support, business transactions and internal services, and processes.
cmdb_ci_information_object	Information Object	Types of information that a business application or any other entity handles. For example: 'Employee Salary Data', Employee Personal Data', and 'Sales Data'.
cmdb_ci_installed_bundles	Installed Bundles	Extension of the Virtual Machine Object [cmdb_ci_vm_object] table, which represents bundles of installed software.

Table name	Table label	Table description
cmdb_ci_infra_service	Infrastructure Service	IT services which support providing computing infrastructure.
cmdb_ci_infra_service_ldap	LDAP Service	Running LDAP service.
cmdb_ci_ip_address	IP Address	IP address.
cmdb_ci_ip_device	IP Device	Base class for devices with an IP address.
cmdb_ci_ip_firewall	IP Firewall	Firewall hardware.
cmdb_ci_ip_network	IP Network	IP network information capturing details such as subnet, router, and router_interface_type.
cmdb_ci_ip_phone	IP Phone	IP-enabled (VOIP) phone.
cmdb_ci_ip_router	IP Router	Specialization of the Network Gear [cmdb_ci_netgear] table.
cmdb_ci_ip_server	IP Server	Server hardware.
cmdb_ci_ip_service	IP Service Instance	Base table for IP services running on a server such Unix daemon or Windows service.
cmdb_ci_ip_switch	IP Switch	Specialization of the Network Gear [cmdb_ci_netgear] table.

Table name	Table label	Table description
cmdb_ci_iplanet_web_server	Iplanet Web Server	Server running Oracle iPlanet Web Server (OiWS) software.
cmdb_ci_isam_server	ISAM Server	Server running ISAM software.
cmdb_ci_iscsi_disk	iSCSI Disk	Host mount of an iSCSI disk.
cmdb_ci_iscsi_export	iSCSI Export	Storage volume exported by a storage server via iSCSI.
cmdb_ci_kubernetes_component	Kubernetes Component	<p>Kubernetes cluster, ingress, namespace, node, pod, service, volume, and workload. Also represents open shift build conf, deployment conf, docker images repository, group, images, images, stream, project, route, and user.</p> <p>Parent class for Kubernetes components such as Pods and Clusters.</p>
cmdb_ci_kvm	KVM	Hypervisor that manages kernel-based virtual machines (KVMs).
cmdb_ci_kvm_network	Network	KVM Virtual network.

Table name	Table label	Table description
cmdb_ci_kvm_object	KVM Object	Base object for all KVM objects.
cmdb_ci_kvm_storage_pool	Storage Pool	KVM storage pool.
cmdb_ci_kvm_storage_volume	Storage Volume	KVM storage volume.
cmdb_ci_kvm_vm_instance	KVM Virtual Machine Instance	Virtual machine instance running on a KVM hypervisor.
cmdb_ci_lb	Load Balancer	Server functioning as a load balancer.
cmdb_ci_lb_a10	A10 Load Balancer	Server functioning as an A10 load balancer.
cmdb_ci_lb_ace	ACE	Server functioning as an ACE load balancer.
cmdb_ci_lb_alteon	Alteon	Server functioning as an Alteon load balancer.
cmdb_ci_lb_appl	Load Balancer Application	Application that provides load balancing functionality.
cmdb_ci_lb_backend_server	LB Backend Server	Server functioning as a backend load balancer.
cmdb_ci_lb_bigip	F5 BIG-IP	Server functioning as an F5 BIG-IP load balancer.

Table name	Table label	Table description
cmdb_ci_lb_cisco_csm	Cisco CSM	Server functioning as a Cisco CSM load balancer.
cmdb_ci_lb_cisco_css	Cisco CSS	Server functioning as a Cisco CSS load balancer.
cmdb_ci_lb_f5_gtm	F5 BigIP GTM	Server functioning as an F5 BigIP GTM load balancer.
cmdb_ci_lb_f5_ltm	F5 BigIP LTM	A server functioning as an F5 BigIP LTM load balancer.
cmdb_ci_lb_haproxy	HAProxy Load Balancer	Server functioning as an HA Proxy load balancer.
cmdb_ci_lb_isa	ISA Server	Server functioning as an ISA load balancer.
cmdb_ci_lb_modjk	Modjk Load Balancer	Server functioning as a Cisco CSM load balancer.
cmdb_ci_lb_modproxy	ModProxy Load Balancer	Server functioning as a ModProxy load balancer.
cmdb_ci_lb_netscaler	Citrix Netscaler	Server functioning as a Citrix Netscaler load balancer.
cmdb_ci_lb_network	Network Load Balancer	Server performing network load balancing.

Table name	Table label	Table description
cmdb_ci_lb_nginx	Nginx Load Balancer	Server functioning as an Nginx load balancer.
cmdb_ci_lb_pool	Load Balancer Pool	Collection of host-to-port mappings to be balanced.
cmdb_ci_lb_pool_member	Load Balancer Pool Member	Host-to-port mapping of a request to be balanced.
cmdb_ci_lb_radware	Radware Load Balancer	Server functioning as a Radware load balancer.
cmdb_ci_lb_service	Load Balancer Service	Virtual service that the device balances by forwarding requests to members within a pool.
cmdb_ci_lb_template	Load Balancer Template	Load balancer template which contains load balancer-related configuration settings for a specific type of network traffic.
cmdb_ci_lif	LIF	Logical interface.
cmdb_ci_lb_vlan	Load Balancer VLAN	Virtual LAN segment.
cmdb_ci_linux_server	Linux Server	Server running Linux software.
cmdb_ci_logical_datacenter	Logical Datacenter	VMware vCenter logical datacenter.

Table name	Table label	Table description
cmdb_ci_lpar	Logical Partition	Logical partition, commonly called an LPAR, is a subset of a computer's hardware resources, virtualized as a separate computer.
cmdb_ci_lvm_pool	LVM Pool	Linux Volume Manager storage pool.
cmdb_ci_lvm_pool_member	LVM Pool Member	Linux Volume Manager storage pool member.
cmdb_ci_mainframe	IBM Mainframe	IBM large-scale computer system.
cmdb_ci_mainframe_hardware	Mainframe Hardware	The hardware components of a large-scale computer system.
cmdb_ci_mainframe_lpar	IBM Mainframe LPAR	Logical partition, which is commonly called an LPAR, and is a subset of a mainframes computer's hardware resources, virtualized as a separate computer.
cmdb_ci_memory_module	Memory Module	Circuit board that provides for memory storage.

Table name	Table label	Table description
cmdb_ci_mfp_printer	Multi-function Printer	Physical device with scan, copy, and fax capabilities.
cmdb_ci_microsoft_iis_web_server	Microsoft IIS Web Server	Server running Internet Information Services (IIS) for Windows software.
cmdb_ci_modem_network	Modem Hardware	Physical modem hardware.
cmdb_ci_mpio_pool	Multipath IO Pool	Multipath IO pool, representing multiple redundant paths to storage.
cmdb_ci_mpio_pool_group	Multipath IO Pool Group	Group of multipath IO pools.
cmdb_ci_mpio_pool_path	Multipath IO Pool Path	Single path in an MPIO pool.
cmdb_ci_msds	Mass Storage Device	Physical storage device.
cmdb_ci_nas_file_system	NAS File System	Extension of the File System [cmdb_ci_file_system] table, representing network attached storage.
cmdb_ci_nat_gateway	NAT Gateway	Functionality for NAT gateway.
cmdb_ci_netapp_cdots	NetApp CDOT	Functionality of NetApp Clustered Data OnTap operating system

Table name	Table label	Table description
cmdb_ci_netapp_datacenter	NetApp Datacenter	NetApp logical datacenter.
cmdb_ci_netapp_svm	NetApp SVM	NetApp Storage Virtual Machine.
cmdb_ci_netapp_volume	NetApp Volume	NetApp FlexVol storage volume.
cmdb_ci_net_app_server	Network Appliance Hardware	Server configured to perform as a networking appliance.
cmdb_ci_netgear	Network Gear	Extension of the Hardware table, that captures network equipment such as router, switch, hub, gateway, and bridge.
cmdb_ci_netware_server	Netware Server	Server running NetWare software.
cmdb_ci_network	Cloud Network	VMware vCenter cloud network.
cmdb_ci_network_acl	Network ACL	Network access control list (ACL).
cmdb_ci_network_acl_rule	Network ACL Rule	Rule used to control networking access rights.
cmdb_ci_network_adapter	Network Adapter	Network adapter hardware.
cmdb_ci_network_policy_group	Network Policy Group	Group policy consumed by Active Directory services.

Table name	Table label	Table description
cmdb_ci_network_template	Network Template	OpenStack file used to configure a network.
cmdb_ci_nic	Cloud Mgmt Network Interface	Virtual network adapter.
cmdb_ci_nginx_web_server	Nginx Web Server	Server running Nginx software.
cmdb_ci_openstack_datacenter	OpenStack Datacenter	OpenStack logical datacenter.
cmdb_ci_optical_transport	Fiber Optic Equipment	<p>Fiber optics which are used for long-distance and high-performance data networking. Fiber optics are commonly used in telecommunication services such as internet, television and telephones.</p> <p>Child class of Transport Hardware and parent class for telecom fiber optic equipment such as optical multiplexers and terminal equipment.</p>
cmdb_ci_oslv_container	Operating-system-level Virtualization Container	Containers (a runtime instance of a docker image).

Table name	Table label	Table description
cmdb_ci_oslv_engine	Operating-system-level Virtualization Engine	Software for running and managing containers.
cmdb_ci_oslv_image	Operating-system-level Virtualization Image	Container images. Ordered collection of root filesystem changes and the corresponding execution parameters for use within a container runtime.
cmdb_ci_oslv_image_tag	Operating-system-level Virtualization Image Tag	Container tag, which is a label applied to a container image in a repository.
cmdb_ci_oslv_local_image	Operating-system-level Virtualization Local Image	Locally managed container image.
cmdb_ci_osx_server	OS/X Server	Server running OS/X operating system.
cmdb_ci_os_template	Image	Software files used to create a new instance of a compute resource such as server, desktop, virtual machine, and virtual router.
cmdb_ci_outofband_device	Out-of-Band Device	Hardware used to perform out-of-band management.
cmdb_ci_patches	Patch	Patch software to fix or improve a

Table name	Table label	Table description
		computer program or its supporting data.
cmdb_ci_pc_hardware	Personal Computer	Multi-purpose electronic computer whose size, capabilities, and price make it feasible for individual use.
cmdb_ci_pcf_component	CloudFoundry Component	<p>Cloud Foundry provides a highly efficient, modern model for cloud native application delivery on top of Kubernetes. Component represents application, domain, organization, quota, routes, service, service plan, space, and space instances.</p> <p>Parent class for price base of cloud products such as virtual machines.</p>
cmdb_ci_pdu	PDU	Power distribution unit (PDU).
cmdb_ci_pdu_outlet	Outlet	Single outlet of a PDU.
cmdb_ci_peripheral	Computer Peripheral	Various computer peripherals such as monitor, docking station, KVM switch, projector, scanner,

Table name	Table label	Table description
		keyboard, mouse, and UPS.
cmdb_ci_personal_printer	Personal Printer	Printer whose size, capabilities and price make it feasible for individual use.
cmdb_ci_plotter	Plotter	Printer with capabilities to print large vector graphic images.
cmdb_ci_power_eq	Power Equipment	Hardware used to manage electrical power.
cmdb_ci_port	Port	Interface between a computer and other electronic devices.
cmdb_ci_port_group	Port Group	Group of ports on a virtual switch.
cmdb_ci_print_queue	Print Queue	Print queue, which is a list of printer output jobs held in a reserved memory area, including the most current status of all active and pending print jobs.
cmdb_ci_printer	Printer	Physical device which makes a persistent human-readable representation of graphics or text on paper or similar physical media.

Table name	Table label	Table description
cmdb_ci_printing_hardware	Printing Hardware	Physical device which makes a persistent human-readable representation of graphics or text on paper or similar physical media.
cmdb_ci_puppet_master	Puppet Primary	Server running PuppetMaster application.
cmdb_ci_qtree	Qtree	Qtree file system.
cmdb_ci_qualifier	Qualifier	<p>Relation qualifier CIs which contain important information about CI relationships.</p> <p>In a relation qualifier, you can annotate arbitrary unique information about the relationship between two CIs. You can define multiple qualifiers for a single relationship, resulting in a qualifier chain. However, there can be only a single qualifier chain for a specific relationship type between two CIs.</p> <p>Parent class for the various relation qualifier types such as EntryPoint Markers</p>

Table name	Table label	Table description
		and Boundary Connections.
cmdb_ci_rack	Rack	Datacenter racks containing details such as rack units, rack units in use, and power consumption.
cmdb_ci_raid	RAID	Storage pool using RAID mechanisms to ensure data integrity.
cmdb_ci_raid_member	RAID Member	Member of storage pool using RAID mechanisms.
cmdb_ci_resource_group	Resource Group	Resource pool is a logical abstraction for flexible management of resources.
cmdb_ci_sa_scaling_pol_base	Scaling Policy	Parent class for virtual machines scaling policy options such as simple and dynamic.
cmdb_ci_san	Storage Area Network	Network which provides access to block level storage.
cmdb_ci_san_connection	SAN Connection	Connection in a SAN network.
cmdb_ci_san_disk	SAN Disk	Base table for the iSCSI Disk [cmdb_ci_iscsi_disk] and the Fibre Channel Disk [cmdb_ci_fc_disk] tables.

Table name	Table label	Table description
cmdb_ci_san_endpoint	SAN Endpoint	One end of a SAN connection.
cmdb_ci_san_export	SAN Export	Base table for the iSCSI Export [cmdb_ci_iscsi_export] and the Fibre Channel Export [cmdb_ci_fc_export] tables.
cmdb_ci_san_fabric	SAN Fabric	Hardware that connects workstations and servers to storage devices in a SAN. Referred to as a "fabric."
cmdb_ci_san_zone	SAN Zone	Subset of SAN storage that certain users are restricted to.
cmdb_ci_san_zone_alias	SAN Zone Alias	Collection of SAN zone members.
cmdb_ci_san_zone_alias_member	SAN Zone Alias Member	M2m relationship between SAN zone aliases and SAN zone members.
cmdb_ci_san_zone_member	SAN Zone Member	Ports and devices in a SAN zone.
cmdb_ci_san_zone_set	SAN Zone Set	Collection of SAN zones.
cmdb_ci_scanner	Scanner	Hardware used to create digital imagine of paper documents.

Table name	Table label	Table description
cmdb_ci_server	Server	Base class for all types of servers.
cmdb_ci_server_snapshot	Server Snapshot	Server snapshot, which is the state of a system at a particular point in time.
cmdb_ci_service	Service	IT Service that directly supports a Business Process (ITIL).
cmdb_ci_service_auto	Application Service	Services that can be monitored by the system, which in the base system, includes only application services. If Service Mapping is activated, there can also be records for dynamic CI groups. If Event Management is activated, there can be records for alert groups.
cmdb_ci_service_business	Business Service	Business services are published to business users and typically underpin one or more business capabilities. Business services are often orderable by business users.
cmdb_ci_service_discovered	Mapped Application Service	Application services, created by the Manual service population method. For each application

Table name	Table label	Table description
		service, there is a container CI record that models the application service.
cmdb_ci_service_technical	Technical Service	Technical services are published to service owners and typically underpin one or more business services. A technical service may have an operational view made up of one or more technical service offerings.
cmdb_ci_solaris_instance	Solaris Virtual Machine Instance	Virtual machine instance running Solaris software.
cmdb_ci_solaris_server	Solaris Server	Physical server running Solaris software.
cmdb_ci_spkg	Software	Software package information containing details such as version, install count, license count, package name, and key (when SAM enabled).
cmdb_ci_storage_controller	Storage Controller	Logical device that controls a storage volume or Fibre Channel port.
cmdb_ci_storage_device	Storage Device	Base table for block storage devices such as DAS, SAN, and NAS.

Table name	Table label	Table description
cmdb_ci_storage_disk	Storage Disk	Disk installed in a storage server.
cmdb_ci_storage_export	Storage Export	Base table for the SAN Export [cmdb_ci_san_export] table.
cmdb_ci_storage_fileshare	Storage File Share	NAS file system on a storage server (an exported file system).
cmdb_ci_storage_hba	Storage HBA	Host bus adapter for Fibre Channel. The physical device that provides Fibre Channel ports.
cmdb_ci_storage_pool	Storage Pool	Logical collection of storage.
cmdb_ci_storage_pool_member	Storage Pool Member	Logical volume in a storage pool.
cmdb_ci_storage_switch	Storage Switch	Fibre Channel switch.
cmdb_ci_storage_volume	Storage Volume	Volume on a storage server.
cmdb_ci_storage_vol_snapshot	Storage Volume Snapshot	Server snapshot is the state of a system at a particular point in time.
cmdb_ci_subnet	Cloud Mgmt Subnet	Part of a larger network.
cmdb_ci_sun_dir_proxy_server	Sun Directory Proxy Server	Server running Sun ONE Directory Proxy Server software.

Table name	Table label	Table description
cmdb_ci_sun_ldap_dir_server	Sun LDAP Server	Server running Sun ONE Directory Server (LDAP) software.
cmdb_ci_surge_power_eq	Surge Protection Equipment	Power equipment used to prevent power surges.
cmdb_ci_tape_server	Server Tape Unit	Hardware for using magnetic tape storage.
cmdb_ci_tomcat_connector	Tomcat Connector	Software which provides web server plugins to connect web servers with Tomcat and other backends.
cmdb_ci_tower_eq	Tower Equipment	Parent class for facility towers and tower equipment such as lights and beacons.
cmdb_ci_translation_rule	NAT	Rules to allow router to remap one network address to another.
cmdb_ci_transport_hardware	Transport Hardware	Telecommunication hardware used for digital communication and related hardware such as telecommunication racks, servers to connect port to LAN, relays, channel banks, and network circuit switch.

Table name	Table label	Table description
		Parent class for telecom transport hardware such as multiplexers and fiber optic equipment.
cmdb_ci_ucs_blade	Cisco UCS Blade	Physical Cisco UCS Blade server hardware.
cmdb_ci_ucs_chassis	Cisco UCS Chassis	Physical Cisco UCS chassis hardware used to hold Cisco UCS Blade server hardware.
cmdb_ci_ucs_equipment	Cisco UCS Equipment	Cisco Unified Computing System (UCS) products.
cmdb_ci_unix_cluster	UNIX Cluster	Set of computers clustered together to present a single Unix server resource.
cmdb_ci_unix_daemon	UNIX Daemon	Long running Unix background process used to answer requests for services.
cmdb_ci_unix_server	UNIX Server	Server running Unix software.
cmdb_ci_ups	UPS	Uninterrupted Power Supply devices, where devices are traditional UPS devices.
cmdb_ci_ups_alarm	UPS Alarm	Uninterrupted Power Supply alarm.

Table name	Table label	Table description
cmdb_ci_ups_bypass	UPS Bypass	Uninterrupted Power Supply bypass.
cmdb_ci_ups_input	UPS Input	Electrical input to an Uninterrupted Power Supply device.
cmdb_ci_ups_output	UPS Output	Electrical output from an Uninterrupted Power Supply device.
cmdb_ci_ups_power_eq	Uninterruptible Power Supply	Uninterrupted Power Supply devices, where devices are any non-traditional UPS devices that manage electrical power.
cmdb_ci_vcenter	VMware vCenter Instance	Installed instance of VMware VCenter software.
cmdb_ci_vcenter_cluster	VMware vCenter Cluster	Set of servers that work together while running VMware VCenter software.
cmdb_ci_vcenter_cluster_drs_rule	VMWare vCenter Cluster DRS Rule	vCenter specific table which stores Distributed Resource Scheduler (DRS) basic rule information. Stores the affinity rule specifying that the members of a selected virtual machine DRS group can or must run on the

Table name	Table label	Table description
		members of a specific host DRS group. Populated by the vCenter API ClusterRuleInfo(vim.cluster.RuleInfo) .
cmdb_ci_vcenter_datacenter	VMware vCenter Datacenter	VMware VCenter data center objects.
cmdb_ci_vcenter_datastore	VMware vCenter Datastore	VMware VCenter datastore objects containing details such as capacity, freespace, filesystem, and type.
cmdb_ci_vcenter_datastore_disk	Datastore Disk	Individual VMware VCenter datastore disk.
cmdb_ci_vcenter_folder	VMware vCenter Folder	VCenter folders, which can be used to group objects of the same type for easier management.
cmdb_ci_vcenter_network	VMware vCenter Network	VMWare vCenter virtual network.
cmdb_ci_vcenter_object	VMware vCenter Object	Base class for most VMware VCenter objects.
cmdb_ci_vcenter_server_obj	VMware vCenter Server Object	VMware hypervisor.

Table name	Table label	Table description
cmdb_ci_veritas_disk	Veritas Disk	Physical disk that is controlled by Veritas Volume Manager.
cmdb_ci_veritas_disk_group	Veritas Disk Group	Collection of disks in Veritas Volume Manager.
cmdb_ci_veritas_plex	Veritas Plex	Logical partition in Veritas Volume Manager.
cmdb_ci_veritas_subdisk	Veritas Subdisk	Partition of a Veritas_disk.
cmdb_ci_veritas_volume	Veritas Volume	Aggregation of plexes.
cmdb_ci_virtualization_server	Virtualization Server	Base table used by the ESX Server [cmdb_ci_esx_server] table.
cmdb_ci_virtual_desktop	Virtual Desktop	User's desktop environment (such as icons, wallpaper, windows, folders, toolbars, and widgets) is stored remotely on a server.
cmdb_ci_virtual_pvt_gateway	Virtual Private Gateway	Two VPN endpoints for automatic failover.
cmdb_ci_vm	Virtual Machine HyperVisor	Hypervisor software.
cmdb_ci_vm_instance	Virtual Machine Instance	Generic virtual machines information.

Table name	Table label	Table description
cmdb_ci_vm_object	Virtual Machine Object	Base class for all VM objects. Parent for all objects such as Hyper-V object and KVM object.
cmdb_ci_vm_parallel	Parallels	Instance of Parallels software.
cmdb_ci_vm_template	Virtual Machine Template	Template, which is a master copy of a virtual machine that can be used to create many clones. Base table for all VM templates.
cmdb_ci_vm_user_credentials	VM User Credentials	Credentials, which are used to authenticate access rights and permissions.
cmdb_ci_vm_vmware	VMware	VMWare specialization of the Virtual Machine table. No longer used.
cmdb_ci_vm_zones	Zones	Partitioned virtual OS environment working in a Solaris operating system space.
cmdb_ci_voice_hardware	Voice System Hardware	Telecommunication systems that represent telephone systems, telephone exchanges, and voice mail. Parent class for telecom voice system hardware such as

Table name	Table label	Table description
		voicemail and private branch exchanges.
cmdb_ci_volume_template	Volume Template	Set of rules that specify one or more capabilities of a storage volume (storage selection and layout rules).
cmdb_ci_vmware_instance	VMware Virtual Machine Instance	VMware VM instance on VMware hypervisor.
cmdb_ci_vmware_template	VMware Virtual Machine Template	Master copy of a VMware virtual machine that can be used to create many clones.
cmdb_ci_vpc	Virtual Private Cloud	On-demand configurable pool of shared computing resources allocated within a public cloud environment.
cmdb_ci_vpn	Virtual Private Network	Private network configured to run across a public network.
cmdb_ci_vpn_connection	VPN Connection	Secure connection to another network over the Internet.
cmdb_ci_vserver_peer	Vserver Peer	Extends the Virtual Machine Object [cmdb_ci_vm_object] table. Peer relationship

Table name	Table label	Table description
		between the Vservers (NetApp).
cmdb_ci_wap_network	Wireless Access Point	Networking hardware device that allows a Wi-Fi compliant device to connect to a wired network.
cmdb_ci_web_application	Web Application	Client–server software application in which the client (or user interface) runs in a web browser.
cmdb_ci_web_domino	Lotus Domino HTTP Server	Server running IBM Notes software (formerly Lotus Notes/IBM Domino).
cmdb_ci_web_server	Web Server	Computer system that processes requests via HTTP from the World Wide Web.
cmdb_ci_web_service	Web Service	Service offered by an electronic device to another electronic device, communicating with each other via the World Wide Web.
cmdb_ci_web_site	Web Site	A collection of related web pages.
cmdb_ci_websphere_cell	Websphere Cell	A logical grouping of IBM Websphere nodes (each of which runs one or more application servers)

Table name	Table label	Table description
		that are centrally managed.
cmdb_ci_win_cluster	Windows Cluster	A single (virtual) server composed of one or more physical Windows Servers.
cmdb_ci_win_cluster_node	Windows Cluster Node	A physical member of the Windows Cluster application.
cmdb_ci_win_cluster_resource	Windows Cluster Resource	A logical or physical entity managed by the Windows Cluster application.
cmdb_ci_win_domain_controller	Windows Domain Controller	A server that responds to security authentication requests within a Windows Server domain.
cmdb_ci_win_server	Windows Server	A server running Microsoft Windows Server operating system.
cmdb_ci_windows_service	Windows Service	A Windows computer program that operates in the background.
cmdb_ci_zone	Data Center Zone	A specified portion of a data center facility.

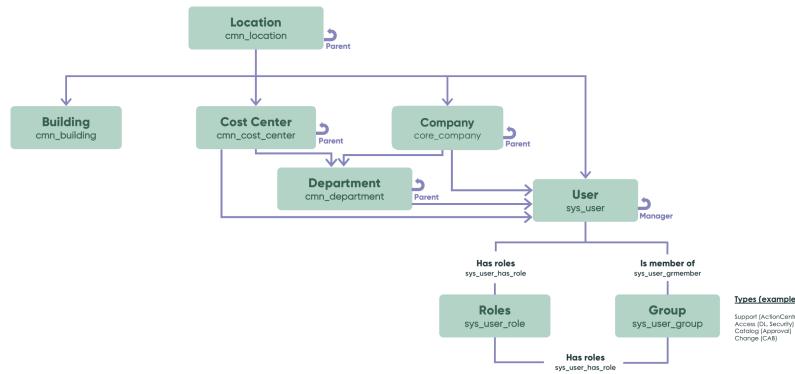
Configuration Item [cmdb_ci] class

Attributes in the Configuration Item [cmdb_ci] class, which extends the Base Configuration Item [cmdb] class.

Warning: Do not modify any of these attributes in the dictionary. For example, do not modify the type of the location attribute from reference to list. Such modifications may prevent features that use the CMDB, from functioning properly.

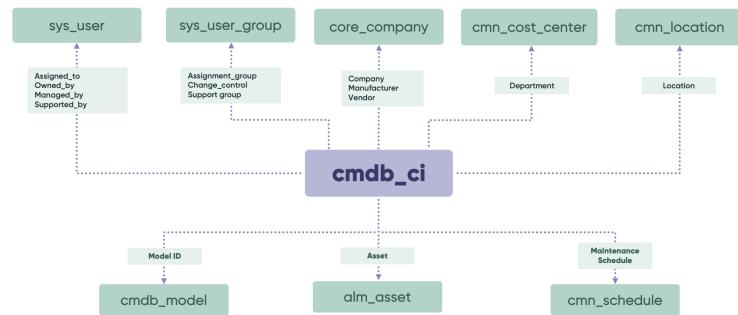
For descriptions of common CMDB tables in a base system, see [CMDB tables descriptions](#).

Common, core, user tables



now.

CMDB CI schema related to common core and non-core tables



Attributes

Attribute	Description
Asset tag	Asset tag/service tag for the specific asset
Assigned	Date and time of assignment to user
Attributes	Description of usage of attributes for the instance
Can Print	Indicates whether the instance can print
Category	Name of category applicable to the instance
Checked in	Date and time of checking in
Checked out	Date and time of checking out

Attribute	Description
Class	System class name
Comments	Comments related to the instance
Correlation ID	ID of the instance from another data source
Cost	Financial value in local currency (as defined in the Cost Currency field)
Cost currency	Name of currency (such as dollars, pounds, Euros)
Created	Date and time record was created
Created by	Name of person/data source which initially created the record
Description	Fit (how deployed) and function (purpose) of the instance
Discovery source	Name of primary (most trusted) discovery source
DNS Domain	Name of the DNS domain to which the instance belongs
Domain	ID of the domain to which the instance belongs
Domain Path	Path of the domain to which the instance belongs
Due	Date and time instance was due
Due in	Description of the manner of which the instance was due

Attribute	Description
Fault Count	Number of faults recorded against the instance to date
First Discovered	Date and time instance was initially discovered
Fully Qualified Domain Name	Full path name of domain to which the instance belongs
GL account	General Ledger account name/number
Installed	Date and time instance was most recently installed
Invoice number	Invoice number used in acquisition process
IP Address	Primary IP address used by the instance
Justification	Description of the justification for the instance
Lease contract	Number of current leasing contracts
MAC Address	MAC address of the instance
Model Number	Manufacturer original model number
Monitor	Indicates whether the instance is monitored
Most Recent Discovery	Date and time instance was last discovered
Name	Name of the CI instance

Attribute	Description
Operational Status	Configurable choice list for current operational states
Order received	Date and time instance was initially received
Ordered	Date and time instance was initially ordered
PO number	Purchase order number used in acquisition process
Purchased	Date instance was purchased
Requires verification	Flag indicating whether verification is required for the instance
Serial number	Serial number of the instance
Skip sync	Flag indicating whether synchronization between Asset Management and CMDB can be skipped
Start Date	Date and time the instance was last started
Status	Configurable choice list with values for current functional states
Subcategory	Name of Subcategory applicable to the instance
Sys ID	ServiceNow Sys ID (GUID)
Tags	Related tags
Updated	Date and time instance was last updated

Attribute	Description
Updated by	Person/data source which last updated the record
Updates	Configurable choice list with values for update states
Warranty expiration	Date current warranty expires

Reference attribute	Reference to
Approval Group	Group table
Asset	Asset table
Assigned to	User table
Change Group	Group table
Company	Company table
Cost center	Cost Center table
Department	Department table
Location	Location table
Maintenance Schedule	Schedule table
Managed by	User table
Manufacturer	Company table
Model ID	Product Model table
Owned by	User table
Schedule	Schedule table (for normal processing)

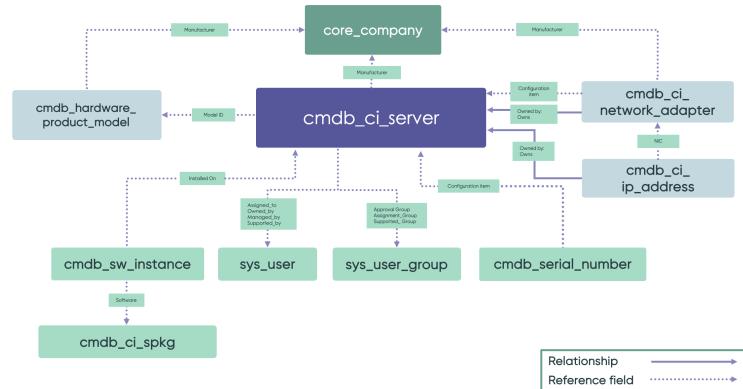
Reference attribute	Reference to
Support group	Group table
Supported by	User
Vendor	Company table

Hardware [cmdb_ci_hardware] class

Attributes, identification rule, and other important schema structures for the CMDB Hardware [cmdb_ci_hardware] class.

For descriptions of common CMDB tables in a base system, see [CMDB tables descriptions](#).

Hardware, Computer, Server Schema



Note: **cmdb_sw_instance** noted in the diagram, is a reference to the **cmdb_software_instance** class.

Attributes

The Hardware class adds the following unique attributes:

Attribute	Description
hardware_status	Status of hardware such as In Maintenance or Retired . Used to sync status to the Asset class.
hardware_substatus	Secondary hardware status. Each setting in hardware_status results in a different set of choices available for this field.
default_gateway	Default gateway that the computer is connected through.

Key relationship structures

Use the following key relationships as important guidelines when creating Hardware, Computer, or Server CIs:

-

Serial number: During CI identification, Identification and Reconciliation Engine (IRE) processes search for a serial number in two locations. One is the CI serial number attribute, and the second one is the Serial Number [cmdb_serial_number] table, with reference back to the Hardware [cmdb_ci_hardware] table.

- Store any serial number of any type other than System, only in the Serial Number table (and not in the server CI attribute).
- If the system serial number is available, store it in both the Serial Number attribute of the CI and in the Serial Number table.

The Serial Number table is a many to one relationship linking back to the server CI. This table has a type field for specifying the type of the serial number (system, uuid, chassis, bios, or baseboard) and storing the actual value itself. Use the following Windows standards for serial number types in the Serial Number [cmdb_serial_number] table:

- system: Product identification such as a serial number for software, a die number on a hardware chip, or a project number (for noncommercial products).

- **uuid:** Universally unique identifier (UUID) for the product. A UUID is a 128-bit identifier that is guaranteed to be different from other generated UUIDs.
- **chassis:** Manufacturer-allocated number that is used to identify a physical element. Value is the Serial Number member of the System Enclosure or Chassis structure in the SMBIOS information. This type represents the properties associated with a physical system enclosure.
- **bios:** The assigned serial number of the BIOS. This type represents the attributes of the computer system basic input/output services (BIOS) that are installed on the computer.
- **baseboard:** Manufacturer-allocated number that is used to identify the physical element. This property is inherited from CIM_PhysicalElement, and is sometimes referred to as the 'Motherboard Serial Number'.

•

Network adapter:

- Use the Network Adapter [cmdb_ci_network_adapter] class to store network adapters.
- Set the Name attribute in the Network Adapter class to be the name of the Network Adapter device (such as eth0, eth1).

•

Set the MAC Address attribute to be the MAC address value. Format the string with colon separators between octets and lower case hexadecimal characters with padded zeros.

For example: 'f8:f2:1e:00:d4:66'

- In the CI Relationship [cmdb_rel_ci] table, create an Owned By::Owns relationship to the associated Hardware CI. Specify a reference from the Network Adapter [cmdb_ci_network_adapter] table using the CI with a reference to the associated Hardware CI.

•

IP address:

- Use the IP Address [cmdb_ci_ip_address] class to store IP addresses.
- Store an IP address value in the IP Address attribute, and in the Name attribute (to avoid empty Name attributes).
- Store an IPv4 IP address value using the format 'NNN.NNN.NNN.NNN', with decimal-based octets and period separators. Non-conforming values should be considered invalid and cleansed to null values.
- Store an IPv6 IP address value using lower case hexadecimal with colon separators. Non-confirming values should be considered invalid and cleansed to null values.
- Set the Netmask attribute to the IP address.
- In the CI Relationship [cmdb_rel_ci] table, create an **Owned By::Owns** relationship to the associated Hardware CI.
- Specify for the IP address a reference to the Network Adapter [cmdb_ci_network_adapter] table using the Configuration Item with a reference to the associated Hardware CI.
- To ensure that base system identification rules work properly, also store the IP address in the associated Network Adapter class.
-

Network adapter and IP address:

- Store the MAC address of the network adapter installed on a server, in the Network Adapter [cmdb_ci_network_adapter] class.
- Store the IP address in the IP Address [cmdb_ci_ip_address] class.
- Do not store the MAC address or the IP address in the Server [cmdb_ci_server] class.

Key reference structures

Use the following key references as important guidelines when creating Hardware, Computer, or Server CIs:

- Software and processes running on a server: The Software [cmdb_ci_spkg] class contains the generic software package that is

related to the server CI. The cmdb_software_instance table instantiates each instance of the software package with:

- One to one reference back to the Server [cmdb_ci_server] class

- Many to one reference back to the Software class

These references are stored in the Installed on and the Product Name reference attributes respectively.

If either the Software Asset Management Foundation [com.snc.sams] or the Software Asset Management [com.snc.software_asset_management] plugin is installed, then store software details in the Software Installation [cmdb_sam_sw_install] table instead of the cmdb_software_instance table.

- The Manufacturer and Model ID are reference attributes to the Company [core_company] and Product Model [cmdb_model] tables respectively.
- The Owned By, Assigned To, Managed By, and Supported By are reference attributes to the User [sys_user] table. The Change Group and Support Group are reference attributes to the Group [sys_user_group] table.

Identification rule

The base system contains pre-defined identification rules for the Hardware, Computer, and Server classes, which are identical. That identification rule has the following key identifier entries, listed in priority order:

1. Identifier entry which uses lookup-based identification specified with Serial Number [cmdb_serial_number] as the lookup table. The Serial Number table is a many to one reference from Serial Number back to the server CI.
2. Identifier entry specified with the Serial Number attribute in the CI.
3. Identifier entry for the Name attribute. If Serial Number is not available, then the Name (which is the hostname) attribute is used. If both the Serial Number and the Name attributes are provided, then Identification and Reconciliation Engine (IRE) looks first for the Serial Number. Then, if a Serial Number is not found, IRE falls back to using Name.

- Identifier entry specified for the MAC Address/IP Address attributes in the Network Adapter table. However, do not rely only on the MAC Address/IP Address.

If both Serial Number and Name are not available, and only MAC Address/IP Address are available, use MAC Address as the name of the CI. Using the MAC Address as the name of the CI ensures that you don't create an empty CI.

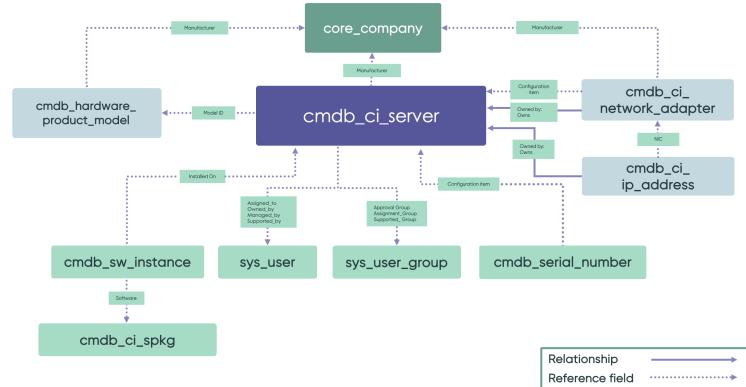
For more information, see [CMDB Identification and Reconciliation](#).

Computer [cmdb_ci_computer] class

Attributes, identification rule, and other important schema structures for the CMDB Computer [cmdb_ci_computer] class.

For descriptions of common CMDB tables in a base system, see [CMDB tables descriptions](#).

Hardware, Computer, Server Schema



Note: **cmdb_sw_instance** noted in the diagram, is a reference to the **cmdb_software_instance** class.

Attributes

The Computer class adds the following unique attributes:

Attribute	Description
CD_ROM	Denotes whether a CD ROM exists.
CD Speed	Speed of CD_ROM.
Chassis type	Type of computer chassis.
CPU core count	Number of cores per CPU.
CPU core thread	Number of threads per core.
CPU count	Number of CPUs.
CPU name	Name of CPU.
CPU speed (MHz)	Speed of CPU.
CPU type	CPU type.
Disk space (GB)	Amount of disk space (in GB).
Floppy	Type of floppy drive.
Form factor	Form factor of the computer.
Object ID	Object ID of the computer (such as the virtual machine ID associated with the computer).
Operating System	Name of the operating system.
OS Address Width (bits)	Operating system bit (such as 32, 64).
OS Domain	NA
OS Service Pack	Service pack installed on the operating system.
OS Version	Version of the operating system.

Attribute	Description
RAM (MB)	Amount of RAM on the computer.
IsVirtual	True/False denoting whether the device is running on a virtual machine instance.

Reference attribute	Reference to
CPU manufacturer	Company [core_company] table

Schema description

The CMDB schema model does not separate between servers and computers that are physical and servers and computer that are virtual. Instead, the Computer and the Server classes have a field named IsVirtual. If a computer or server is a virtual instance, set this IsVirtual attribute to **true**.

As described in the Virtual Machine schema section, to fully model a virtual machine, set the IsVirtual attribute in the Computer or Server CI to **true**. Create a Virtual Machine Instance [cmdb_ci_vm_instance] record using the BIOS UUID attribute as the key identifier. Then create a Hosted_On relationship between the Virtual Machine Instance record and the Computer or Server instance and set IsVirtual to **true**.

Key relationship structures

Use the following key relationships as important guidelines when creating Hardware, Computer, or Server CIs:

-

Serial number: During CI identification, Identification and Reconciliation Engine (IRE) processes search for a serial number in two locations. One is the CI serial number attribute, and the second one is the Serial Number [cmdb_serial_number] table, with reference back to the Hardware [cmdb_ci_hardware] table.

- Store any serial number of any type other than System, only in the Serial Number table (and not in the server CI attribute).

- If the system serial number is available, store it in both the Serial Number attribute of the CI and in the Serial Number table.

The Serial Number table is a many to one relationship linking back to the server CI. This table has a type field for specifying the type of the serial number (system, uuid, chassis, bios, or baseboard) and storing the actual value itself. Use the following Windows standards for serial number types in the Serial Number [cmdb_serial_number] table:

- system: Product identification such as a serial number for software, a die number on a hardware chip, or a project number (for noncommercial products).
- uuid: Universally unique identifier (UUID) for the product. A UUID is a 128-bit identifier that is guaranteed to be different from other generated UUIDs.
- chassis: Manufacturer-allocated number that is used to identify a physical element. Value is the Serial Number member of the System Enclosure or Chassis structure in the SMBIOS information. This type represents the properties associated with a physical system enclosure.
- bios: The assigned serial number of the BIOS. This type represents the attributes of the computer system basic input/output services (BIOS) that are installed on the computer.
- baseboard: Manufacturer-allocated number that is used to identify the physical element. This property is inherited from CIM_PhysicalElement, and is sometimes referred to as the 'Motherboard Serial Number'.

•

Network adapter:

- Use the Network Adapter [cmdb_ci_network_adapter] class to store network adapters.
 - Set the Name attribute in the Network Adapter class to be the name of the Network Adapter device (such as eth0, eth1).
-

Set the MAC Address attribute to be the MAC address value. Format the string with colon separators between octets and lower case hexadecimal characters with padded zeros.

For example: 'f8:f2:1e:00:d4:66'

- In the CI Relationship [cmdb_rel_ci] table, create an Owned By::Owns relationship to the associated Hardware CI. Specify a reference from the Network Adapter [cmdb_ci_network_adapter] table using the CI with a reference to the associated Hardware CI.

•

IP address:

- Use the IP Address [cmdb_ci_ip_address] class to store IP addresses.
- Store an IP address value in the IP Address attribute, and in the Name attribute (to avoid empty Name attributes).
- Store an IPv4 IP address value using the format 'NNN.NNN.NNN.NNN', with decimal-based octets and period separators. Non-conforming values should be considered invalid and cleansed to null values.
- Store an IPv6 IP address value using lower case hexadecimal with colon separators. Non-confirming values should be considered invalid and cleansed to null values.
- Set the Netmask attribute to the IP address.
- In the CI Relationship [cmdb_rel_ci] table, create an **Owned By::Owns** relationship to the associated Hardware CI.
- Specify for the IP address a reference to the Network Adapter [cmdb_ci_network_adapter] table using the Configuration Item with a reference to the associated Hardware CI.
- To ensure that base system identification rules work properly, also store the IP address in the associated Network Adapter class.

•

Network adapter and IP address:

- Store the MAC address of the network adapter installed on a server, in the Network Adapter [cmdb_ci_network_adapter] class.
- Store the IP address in the IP Address [cmdb_ci_ip_address] class.
- Do not store the MAC address or the IP address in the Server [cmdb_ci_server] class.

Key reference structures

Use the following key references as important guidelines when creating Hardware, Computer, or Server Cls:

- Software and processes running on a server: The Software [cmdb_ci_spkg] class contains the generic software package that is related to the server CI. The cmdb_software_instance table instantiates each instance of the software package with:
 - One to one reference back to the Server [cmdb_ci_server] class
 - Many to one reference back to the Software classThese references are stored in the Installed on and the Product Name reference attributes respectively.

If either the Software Asset Management Foundation [com.snc.sams] or the Software Asset Management [com.snc.software_asset_management] plugin is installed, then store software details in the Software Installation [cmdb_sam_sw_install] table instead of the cmdb_software_instance table.

- The Manufacturer and Model ID are reference attributes to the Company [core_company] and Product Model [cmdb_model] tables respectively.
- The Owned By, Assigned To, Managed By, and Supported By are reference attributes to the User [sys_user] table. The Change Group and Support Group are reference attributes to the Group [sys_user_group] table.

Identification rule

The base system contains pre-defined identification rules for the Hardware, Computer, and Server classes, which are identical. That

identification rule has the following key identifier entries, listed in priority order:

1. Identifier entry which uses lookup-based identification specified with Serial Number [cmdb_serial_number] as the lookup table. The Serial Number table is a many to one reference from Serial Number back to the server CI.
2. Identifier entry specified with the Serial Number attribute in the CI.
3. Identifier entry for the Name attribute. If Serial Number is not available, then the Name (which is the hostname) attribute is used. If both the Serial Number and the Name attributes are provided, then Identification and Reconciliation Engine (IRE) looks first for the Serial Number. Then, if a Serial Number is not found, IRE falls back to using Name.
4. Identifier entry specified for the MAC Address/IP Address attributes in the Network Adapter table. However, do not rely only on the MAC Address/IP Address.

If both Serial Number and Name are not available, and only MAC Address/IP Address are available, use MAC Address as the name of the CI. Using the MAC Address as the name of the CI ensures that you don't create an empty CI.

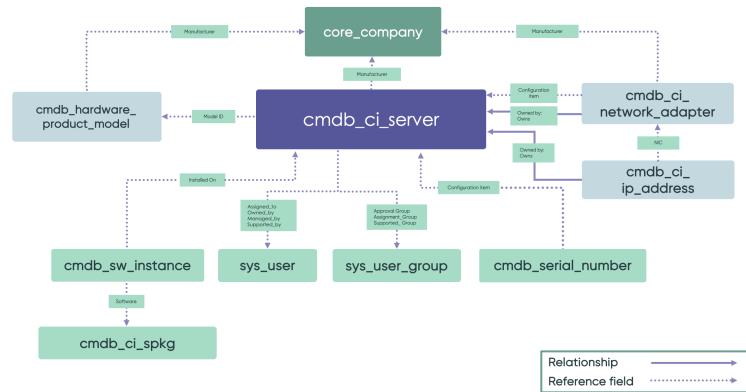
For more information, see [CMDB Identification and Reconciliation](#).

Server [cmdb_ci_server] class

Attributes, identification rule, and other important schema structures for the CMDB Server [cmdb_ci_server] class.

For descriptions of common CMDB tables in a base system, see [CMDB tables descriptions](#).

Hardware, Computer, Server Schema



Attributes

The Server class adds the following unique attributes:

Attribute	Description
Classification	Type of server, such as production, development, disaster recovery, or user acceptance testing (UAT).
Firewall status	Internet or intranet facing server.
Host name	Use the Name attribute to store the host name of the server instead of the Host name attribute.
Used for	Business service supported by the server, such as production, staging, or quality assurance (QA). This attribute uses the Used for

Attribute	Description
	choice list field from the Service [cmdb_ci_service] table.

Reference attribute	Reference to
Disaster backup	Server [cmdb_ci_server] table. Reference to another server that is the backup server for this server.

Reference classes

The following reference classes extend the Server class. They do not add any new attributes.

Reference class	Name	Description
cmdb_ci_solaris_server	Solaris Server	Server running Oracle Solaris operating system.
cmdb_ci_lb_cisco_cm	Cisco CSM	Cisco Security Manager (CSM) load balancer.
cmdb_ci_win_server	Windows Server	Server running Microsoft Windows Server operating system.
cmdb_ci_lb_ace	ACE	Cisco Application Control Engine load balancer.
cmdb_ci_lb_netscaler	Citrix Netscaler	Citrix Netscaler load balancer.
cmdb_ci_lb_alteon	Alteon	Alteon load balancer.

Reference class	Name	Description
cmdb_ci_lb	Load Balancer	Generic load balancer.
cmdb_ci_lb_a10	A10 Load Balancer	A10 load balancer.
cmdb_ci_lb_cisco_css	Cisco CSS	Cisco Content Services Switch (CSS) load balancer.
cmdb_ci_lb_cisco_gss	Cisco GSS	Cisco Global Site Selector (GSS) load balancer.
cmdb_ci_osx_Server	OS/X Server	Server running OS/X operating system.
cmdb_ci_HPUX_Server	HPUX Server	Server running HP-UX operating system.
cmdb_ci_tape_server	Server Tape Unit	Server using a tape drive.
cmdb_ci_Server_Hardware	Server Hardware	Server hardware.
cmdb_ci_datapower_server	Data Power Hosting Server	IBM DataPower hosting server.
cmdb_ci_net_app_server	Network Appliance Hardware	NetApp hardware.
cmdb_ci_netware_server	Netware Server	Server running NetWare operating system.
cmdb_ci_ibm_zos_server	IBM zOS Server	Server running IBM z/OS operating system.

Reference class	Name	Description
cmdb_ci_storage_node_element	Storage Node Element	Storage node.
cmdb_ci_chassis_server	Server Chassis	Server chassis.
cmdb_ci_lb_network	Network Load Balancer	Network load balancer hardware.
cmdb_ci_Unix_Server	Unix Server	Server running Unix operating system.
cmdb_ci_linux_server	Linux Server	Server running Linux operating system.
cmdb_ci_virtualization_server	Virtualization Server	Abstract base table used by Hyper-V Server [cmdb_ci_hyper_v_server] and ESX Server [cmdb_ci_esx_server].
cmdb_ci_mainframe	IBM Mainframe	Large-scale computer system with high-end capabilities.

Schema description

The CMDB schema model does not separate between servers and computers that are physical and servers and computer that are virtual. Instead, the Computer and the Server classes have a field named IsVirtual. If a computer or server is a virtual instance, set this IsVirtual attribute to **true**.

As described in the Virtual Machine schema section, to fully model a virtual machine, set the IsVirtual attribute in the Computer or Server CI to **true**. Create a Virtual Machine Instance [cmdb_ci_vm_instance] record using the BIOS UUID attribute as the key identifier. Then create a Hosted_On relationship between the Virtual Machine Instance record and the Computer or Server instance and set IsVirtual to **true**.

Key relationship structures

Use the following key relationships as important guidelines when creating Hardware, Computer, or Server Cls:

-

Serial number: During CI identification, Identification and Reconciliation Engine (IRE) processes search for a serial number in two locations. One is the CI serial number attribute, and the second one is the Serial Number [cmdb_serial_number] table, with reference back to the Hardware [cmdb_ci_hardware] table.

- Store any serial number of any type other than System, only in the Serial Number table (and not in the server CI attribute).
- If the system serial number is available, store it in both the Serial Number attribute of the CI and in the Serial Number table.

The Serial Number table is a many to one relationship linking back to the server CI. This table has a type field for specifying the type of the serial number (system, uuid, chassis, bios, or baseboard) and storing the actual value itself. Use the following Windows standards for serial number types in the Serial Number [cmdb_serial_number] table:

- system: Product identification such as a serial number for software, a die number on a hardware chip, or a project number (for noncommercial products).
- uuid: Universally unique identifier (UUID) for the product. A UUID is a 128-bit identifier that is guaranteed to be different from other generated UUIDs.
- chassis: Manufacturer-allocated number that is used to identify a physical element. Value is the Serial Number member of the System Enclosure or Chassis structure in the SMBIOS information. This type represents the properties associated with a physical system enclosure.
- bios: The assigned serial number of the BIOS. This type represents the attributes of the computer system basic input/output services (BIOS) that are installed on the computer.
- baseboard: Manufacturer-allocated number that is used to identify the physical element. This property is inherited from

CIM_PhysicalElement, and is sometimes referred to as the 'Motherboard Serial Number'.

-

Network adapter:

- Use the Network Adapter [cmdb_ci_network_adapter] class to store network adapters.
- Set the Name attribute in the Network Adapter class to be the name of the Network Adapter device (such as eth0, eth1).

-

Set the MAC Address attribute to be the MAC address value. Format the string with colon separators between octets and lower case hexadecimal characters with padded zeros.

For example: 'f8:f2:1e:00:d4:66'

- In the CI Relationship [cmdb_rel_ci] table, create an Owned By::Owns relationship to the associated Hardware CI. Specify a reference from the Network Adapter [cmdb_ci_network_adapter] table using the CI with a reference to the associated Hardware CI.

-

IP address:

- Use the IP Address [cmdb_ci_ip_address] class to store IP addresses.
- Store an IP address value in the IP Address attribute, and in the Name attribute (to avoid empty Name attributes).
- Store an IPv4 IP address value using the format 'NNN.NNN.NNN.NNN', with decimal-based octets and period separators. Non-conforming values should be considered invalid and cleansed to null values.
- Store an IPv6 IP address value using lower case hexadecimal with colon separators. Non-confirming values should be considered invalid and cleansed to null values.
- Set the Netmask attribute to the IP address.

- In the CI Relationship [cmdb_rel_ci] table, create an **Owned By::Owns** relationship to the associated Hardware CI.
- Specify for the IP address a reference to the Network Adapter [cmdb_ci_network_adapter] table using the Configuration Item with a reference to the associated Hardware CI.
- To ensure that base system identification rules work properly, also store the IP address in the associated Network Adapter class.
-

Network adapter and IP address:

- Store the MAC address of the network adapter installed on a server, in the Network Adapter [cmdb_ci_network_adapter] class.
- Store the IP address in the IP Address [cmdb_ci_ip_address] class.
- Do not store the MAC address or the IP address in the Server [cmdb_ci_server] class.

Key reference structures

Use the following key references as important guidelines when creating Hardware, Computer, or Server CIs:

- Software and processes running on a server: The Software [cmdb_ci_spkg] class contains the generic software package that is related to the server CI. The cmdb_software_instance table instantiates each instance of the software package with:
 - One to one reference back to the Server [cmdb_ci_server] class
 - Many to one reference back to the Software classThese references are stored in the Installed on and the Product Name reference attributes respectively.

If either the Software Asset Management Foundation [com.snc.sams] or the Software Asset Management [com.snc.software_asset_management] plugin is installed, then store software details in the Software Installation [cmdb_sam_sw_install] table instead of the cmdb_software_instance table.

- The Manufacturer and Model ID are reference attributes to the Company [core_company] and Product Model [cmdb_model] tables respectively.
- The Owned By, Assigned To, Managed By, and Supported By are reference attributes to the User [sys_user] table. The Change Group and Support Group are reference attributes to the Group [sys_user_group] table.

Identification rule

The base system contains pre-defined identification rules for the Hardware, Computer, and Server classes, which are identical. That identification rule has the following key identifier entries, listed in priority order:

1. Identifier entry which uses lookup-based identification specified with Serial Number [cmdb_serial_number] as the lookup table. The Serial Number table is a many to one reference from Serial Number back to the server CI.
2. Identifier entry specified with the Serial Number attribute in the CI.
3. Identifier entry for the Name attribute. If Serial Number is not available, then the Name (which is the hostname) attribute is used. If both the Serial Number and the Name attributes are provided, then Identification and Reconciliation Engine (IRE) looks first for the Serial Number. Then, if a Serial Number is not found, IRE falls back to using Name.
4. Identifier entry specified for the MAC Address/IP Address attributes in the Network Adapter table. However, do not rely only on the MAC Address/IP Address.

If both Serial Number and Name are not available, and only MAC Address/IP Address are available, use MAC Address as the name of the CI. Using the MAC Address as the name of the CI ensures that you don't create an empty CI.

For more information, see [CMDB Identification and Reconciliation](#).

VMware vCenter Object [cmdb_ci_vcenter_object] class

Attributes, identification rule, and other important schema structures for Virtual Machine related classes.

For descriptions of common CMDB tables in a base system, see [CMDB tables descriptions](#).

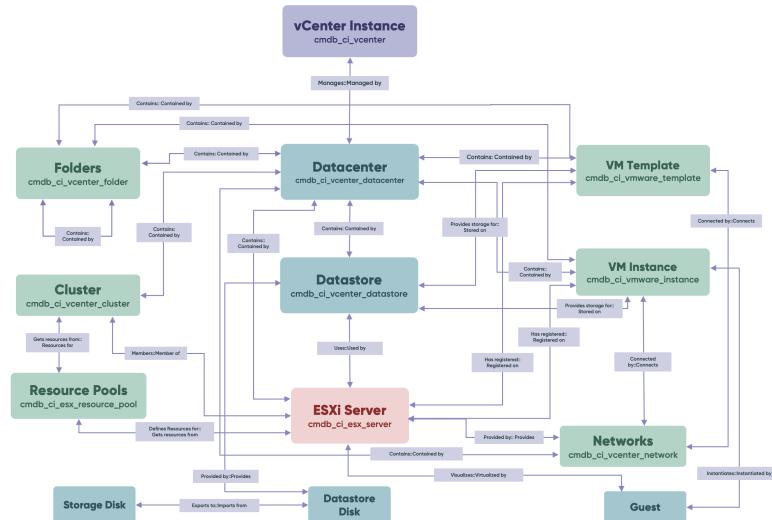
Schema description

ServiceNow® has an extensive modeling of virtual machines (VMs) environment, with classes such as:

- VMware vCenter Cluster [cmdb_ci_vcenter_cluster]
- VMware vCenter datacenter [cmdb_ci_vcenter_datacenter]
- VMware Virtual Machine Instance [cmdb_ci_vmware_instance]

Virtual machines are modeled just like any other server, but with the IsVirtual attribute set to **true**.

VMWare vCenter Instance schema structure



In the diagram above, the 'Discovered' virtual server is referred to as the 'Guest' (VM object). Follow the preceding diagram for any further modeling of VMWare components.

Key reference structures

The Guest has the following important key reference structures:

- An Instantiates::InstantiatedBy relationship with the cmdb_ci_vmware_instance (which is the VM instance reported by Center).
- A Virtualizes::Virtualized by relationship with ESXi Server (the hardware with the ESXi virtualization software installed).
- Guest (Discovered, VMOBJECT) also has a HasRegistered::RegisteredOn relationship to ESXi Server.

Identification rules

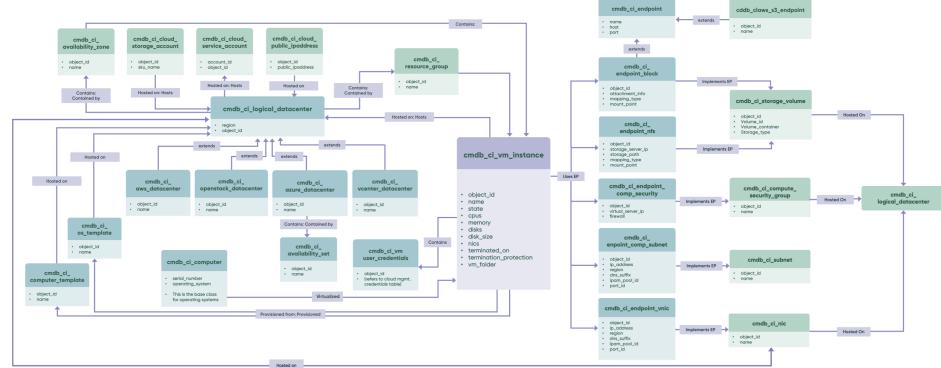
- Guest operating system: Guest operating system is modeled as Server (with IsVirtual set to **true**) and therefore identification rules follow the rules for the Server class. In most operating systems, BIOS UUID is reported as serial number. It is essential that you follow proper VMware guidelines to ensure that BIOS UUID is not being reused. Having a cloned BIOS UUID causes issues with identification rules.
- VM Instances: IRE uses instance MosRef ID as key identifier.
- ESX Server: Server is modeled as bare metal Server (with IsVirtual set to **false**) and therefore applies the Server class identification rules.

Cloud class

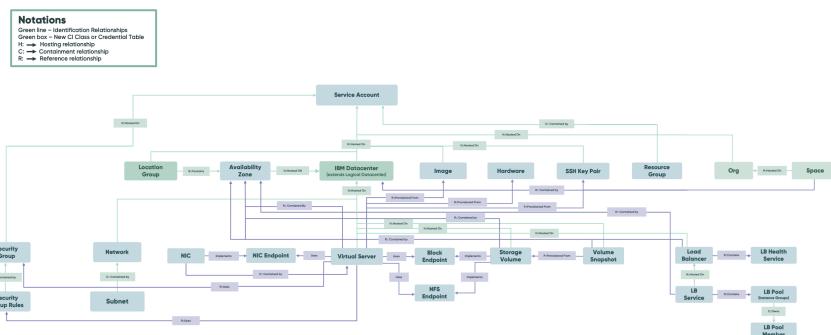
Description, identification rule, and other important schema structures for the CMDB cloud classes.

For descriptions of common CMDB tables in a base system, see [CMDB tables descriptions](#).

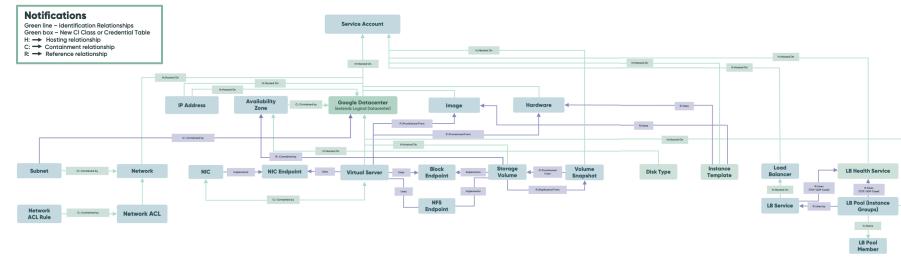
AWS/Azure/OpenStack class model



IBM Datacenter Cloud Schema model



Google Datacenter Schema Model



Cloud schema description

ServiceNow has extensive models of cloud environments including Amazon Web Services (AWS), Microsoft Azure service, Google Cloud Platform (GCP), and IBM Cloud. Focusing on the compute side, the models for cloud environments and for Virtual Servers are similar. For example, instances of Amazon Elastic Compute Cloud (EC2) and Microsoft® Azure Cloud Compute, are an extension of Virtual Machine instances, where CIs are typically created by connecting directly to cloud inventory. However, Virtual Machine instances do not represent actual usage of the cloud instance.

Cloud Service Account [cmdb_ci_cloud_service_account] is the main class for tracking cloud accounts such as AWS, GCP, and Azur (replacing for example, use of the cmdb_ci_aws_account table for AWS).

For example, you can represent a Linux guest host running on Amazon EC2 by the Server [cmdb_ci_server] class, with the IsVirtual attribute set to **true** and with the relationship Runs on:Runs to the EC2 instance. Integrating the AWS Config Service or the Amazon CloudWatch application, provides information on the EC2 object ID. Running Discovery or another discovery program on the guest Linux host, provides the hostname.

Ensure the following:

- Getting the correct UUID which gets stored in the Serial Number [cmdb_serial_number] table.
- Connecting/creating the cloud instance to Host OS, matching on the UUID/Object ID and creating the Runs On:Runs relationship.

Also, there is a complete model of Storage, Networking, Lamda/Functions in addition to modeling of different regions using the concept of the table Logical Datacenter [cmdb_ci_logical_datacenter] with Hosts:HostedOn relationship with Compute, Storage, and such.

Identification rule

The base system contains pre-defined identification rules for cloud schema classes. A cloud object requires the following identification items:

- Object ID: Which is synonymous with the IDs that cloud vendors use for each type of cloud resource, such as Azure Compute, EC2, and Amazon Simple Storage Service (S3).
-

Object ID is unique per region and therefore has dependent relationship requiring information from the Logical Datacenter [cmdb_ci_logical_data_center] table, about the region where the cloud resource is being hosted. For example, AWS Datacenter [cmdb_ci_aws_datacenter], Azure Datacenter [cmdb_ci_azure_datacenter], Google Datacenter [cmdb_ci_google_datacenter] that are extended from Logical Datacenter.

Logical Datacenter itself, has two identifier entries:

- Object ID: Unique ID of the logical datacenter where applicable
- Region: The region of the cloud resource
-

Logical Datacenter has a dependency on cloud service accounts, which has two identifier entries:

- Object ID: Unique ID of the account where applicable.

- Account ID: The unique Account ID that encompasses the different cloud resources. Account ID is generally more applicable than Object ID.

For more information, see [CMDB Identification and Reconciliation](#).

CMDB CI Class Models store app

The ServiceNow Configuration Management Database (CMDB) contains out-of-the-box classes that store data about Configuration Items (CIs). The CMDB CI Class Models store app adds class models that extend the CMDB class hierarchy, including class descriptions, identification rules, identifier entries, and dependent relationships if applicable.

You can use the added classes as any other CMDB class. Applications such as Discovery and Service Mapping can use these class extensions to populate CIs and discover various technologies and software.

Related ServiceNow® Store apps and reference information:

- [CMDB schema model](#): A collection of class diagrams and class attributes for key CMDB classes.
- [CMDB tables descriptions](#): Descriptions of key CMDB tables in the base system.
- [Populating the CMDB](#): Information about the various options for populating the CMDB.
- [Discovery patterns](#): A ServiceNow Store app that provides a library of Discovery patterns for discovering specific devices and applications in the industry.
- [Service Graph Connectors](#): ServiceNow Store apps that provide pre-defined integrations for importing and integrating common third-party data into CMDB classes. Also includes the [IntegrationHub ETL](#) wizard for creating new ETL transform maps.

Add class models

The app adds classes, columns, and the associated metadata as related records in the following tables:

- CMDB Class Information [cmdb_class_info]: Class descriptions

- Identifier [cmdb_identifier]: Identification rules
- Identifier Entry [cmdb_identifier_entry]: Identification entries
- CMDB Metadata Hosting Rules [cmdb_metadata_hosting]: Dependent relationships

Discover using extension classes

The table lists the software and technologies that applications can discover using the extension classes. It provides links to documentation for CMDB CI Class Models and the corresponding ServiceNow Store discovery patterns.

Software/Technology	CMDB CI Class Models Store app	ServiceNow Store discovery patterns
Avi load balancer	Avi load balancer extension classes	Avi Vantage load balancer discovery
BYOL Model of RDS for Oracle	BYOL model of RDS for Oracle extension classes	
Firewall	Firewall extension classes	
IBM Hardware Management Console (HMC)	IBM Hardware Management Console (HMC) extension classes	IBM Virtualization and Hardware Management Console discovery
Internet of Things (IoT)	Internet of Things (IoT) extension classes	N/A
Nutanix	Nutanix extension classes	Nutanix Acropolis discovery
OpenStack	OpenStack extension classes	OpenStack resource discovery

Software/Technology	CMDB CI Class Models Store app	ServiceNow Store discovery patterns
Red Hat Virtualization (RHV)	Red Hat Virtualization (RHV) extension classes	Red Hat Virtualization discovery
Transport Layer Security (TLS)	Transport Layer Security (TLS) extension classes	Discovery procedures provided by Certificate Inventory and Management ServiceNow Store app
VMware NSX load balancer	VMware NSX load balancer extension classes	VMware NSX Advanced load balancer discovery

Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

Verify successful installation

After installing the CMDB CI Class Models store app, make sure the classes were added successfully:

1. Navigate to **All > Configuration > CI Class Manager**.
2. Click **Hierarchy** to display the CI Classes list.

This list contains the added classes, such as the Nutanix classes.

3. Select a class to see the corresponding class details, identification rules, identifier entries, and dependent relationships, if applicable.

Note: Uninstalling the CMDB CI Class Models application might compromise the integrity of the CMDB and result in unexpected behavior.

API extension classes

The CMDB CI Class Models store app adds or updates classes for APIs (application programming interface).

The app adds class models that extend the CMDB class hierarchy, including class descriptions, identification rules, identifier entries, and dependent relationships (if applicable). You can use the added classes as any other CMDB class. Applications such as Discovery and Service Mapping patterns can use these class extensions to populate CIs and discover various technologies and software.

Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

APIs

APIs, are a set of definitions and protocols that enable computer programs to communicate with each other, which enables you to build or integrate application software. APIs typically use web-based technology to communicate with other APIs. APIs are generally used to programmatically perform jobs or tasks, or to view, import, export, delete, or modify data.

The classes added in this release extend the data model and provide a foundation for the representation of API CI classes. You can use this foundation to:

- Gain greater visibility into your APIs.
- Identify security issues and vulnerabilities associated with an API endpoint.

Classes

This section lists the classes that the CMDB CI Class Models store app adds or updates.

CMDB CI Class Models: Release 1.49.0 adds the following classes for API. For the list of CMDB classes in a base system, including ones that this store app might be extending, see [CMDB tables descriptions](#).

Class	Extends	Description
API [cmdb_ci_api]	Configuration Item [cmdb_ci]	APIs that enable two computer programs to communicate with each other, typically using web-based technologies. Example: ChatAPI (https://[apiID].execute-api.us-east-2.amazonaws.com).
API Component [cmdb_ci_api_component]	Configuration Item [cmdb_ci]	Reusable objects related to your API definition that facilitate functionality or exchange of data. Example: GET https://[instance].service-now.com/api/now/table/{tablename} .
API Frontend [cmdb_ci_api_frontend]	API Component [cmdb_ci_api_component]	The part of an API from which a client or user interacts or makes requests. Example: GET https://[apiID].execute-api.us-east-2.amazonaws.com/{proxy+}
API Backend	API Component [cmdb_ci_api_component]	The part of an API that fulfills requests by interacting with backend services, such

Class	Extends	Description
[cmdb_ci_api_backend]		as servers. Example: Lambda:Chat-API-Proxy.
API Gateway [cmdb_ci_api_gateway]	Application [cmdb_ci_appl]	API infrastructure that centralizes client API requests and manages backend processes and services. Example: Kong Gateway.
Managed API [cmdb_ci_managed_api]	Configuration Item [cmdb_ci_api]	API discovered from a gateway or management service. You can enforce a dependency on a gateway for APIs in this class.

Class attributes

CMDB CI Class Models: Release 1.49.0 adds the following attributes to the respective classes.

API [cmdb_ci_api]

Attribute	Data type	Description
Base URL	String (1024)	Base address from which all API components extend.
ID	String (1024)	Unique identifier from the source system.
Type	Choice list	Type of API. You can specify: <ul style="list-style-type: none">• REST

Attribute	Data type	Description
		<ul style="list-style-type: none"> • SOAP • HTTP • gRPC • GraphQL • Websocket
Version	Numeric	Version of the API.
Spec Location	URL	URL to the location of the API specification. Example: OpenAPI spec definition.

API Component [cmdb_ci_api_component]

Attribute	Data type	Description
Method	String	REST API methods. Examples: <ul style="list-style-type: none"> • GET • POST • DELETE
Protocol	String	Communication protocol. Example: HTTP, HTTPS.
Host	String (100)	System that hosts the API.
Path	String (1024)	Specific route the API follows.

Attribute	Data type	Description
Port	String	Communication port. Example: 80, 443, etc.
URL	String (1024)	URL of the resource being called.
ID	String (1024)	Unique identifier from the source system.
Internet Facing	Boolean	Boolean that denotes whether the component is reachable from the public internet. Specify 1 or "true" if the component is reachable.
Authorization	String	Type of authorization or authentication method. Example: <ul style="list-style-type: none"> • Basic • Key • Oauth • None
Request data types	String (255)	List of data types in the request. Examples: <ul style="list-style-type: none"> • CC • Email • Address

Attribute	Data type	Description
Response data types	String (255)	<p>List of data types in the response.</p> <p>Examples:</p> <ul style="list-style-type: none"> • CC • Email • Address

API Frontend [cmdb_ci_api_frontend]

Attribute	Data type	Description
Parent ID	Reference to [cmdb_ci_api_frontend]	Reference to a parent API component.

API Backend [cmdb_ci_api_backend]

Attribute	Data type	Description
Type	String	<p>Backend protocol types of the API.</p> <p>Examples:</p> <ul style="list-style-type: none"> • Lambda • HTTP • Logic App

API Gateway [cmdb_ci_api_gateway]

Attribute	Data type	Description
ID	String (255)	Unique identifier from the source system.

Note: Managed API [cmdb_ci_managed_api] is specific to APIs discovered from gateways and other managed services, and does not introduce new attributes at this time.

Key Relationship Structures

There are a number of key relationships that need to be defined for API and Kong classes.

API relationships

Parent class	Relationship	Child class	Relationship type
API [cmdb_ci_api]	Uses::Used by	API Component [cmdb_ci_api_component]	Suggested
API Gateway [cmdb_ci_api_gateway]	Provides::Provided By	Managed API [cmdb_ci_managed_api]	Dependent
API Frontend [cmdb_ci_api_frontend]	Use End Point To::Use End Point From	API Backend [cmdb_ci_api_backend]	Suggested
API Backend [cmdb_ci_api_backend]	Uses::Used By	Kong Load Balancer [cmdb_ci_kong_lb]	Suggested

Related non-CMDB tables

CMDB CI Class Models v 1.49.0 introduces these non-CMDB tables as related lists for the following API extension classes:

API related list**API Deployment [api_deployment]**

Attribute	Data type	Description
Name	String (100)	Name of the API deployment.
API	Reference	Reference to the deployed API (cmdb_ci_api).
Unmatched API Endpoint	Reference	Reference to the unmatched API endpoint, if the API doesn't match an existing API (cmdb_ci_unmatched_api_endpoint)
Configuration Item	Reference	Reference to the Configuration Item. This is typically manually specified as a reference, if you know what CI the API is deployed to.

Note: The API Deployment non-CMDB table relates to both the API [cmdb_ci_api] and Unmatched API Endpoint [cmdb_ci_unmatched_api_endpoint] classes.

API Component related list**API Header [api_header]**

Attribute	Data type	Description
Name	String (100)	Name of the API header.

Attribute	Data type	Description
API Component	Reference	Reference to the component where the API header is defined (cmdb_ci_api_component).
Unmatched API Endpoint	Reference	Reference to the unmatched API when the endpoint can't be matched to an existing API or component (cmdb_ci_unmatched_endpoint).

API Gateway related lists

API Consumer [api_consumer]

Attribute	Data type	Description
Username	String (100)	Name of the API consumer.
ID	String (255)	Unique identifier from the source system.
Custom ID	String (100)	Alternate display name of the user.
API Gateway	Reference	Reference to the gateway where the consumer is defined (cmdb_ci_api_gateway).

API Policy [api_policy]

Attribute	Data type	Description
Name	String (100)	Name of the API policy.
ID	String (255)	Unique identifier from the source system.
Frontend	Reference	Reference to the API Frontend (cmdb_ci_api_frontend).
Managed API	Reference	Reference to the Managed API (cmdb_ci_managed_api).
Consumer	Reference	Reference to the API Consumer (api_consumer) non-CMDB table.
Protocols	String	Array of protocols that this API policy can apply to.
Active	Boolean	Determines if this non-CMDB table is considered active or inactive.
API Gateway	Reference	Reference to the gateway where the consumer is defined (cmdb_ci_api_gateway).

The CMDB CI Class Models store app adds or updates classes for unmatched APIs (application programming interface).

The app adds class models that extend the CMDB class hierarchy, including class descriptions, identification rules, identifier entries, and dependent relationships (if applicable). You can use the added classes as any other CMDB class. Applications such as Discovery and Service Mapping patterns can use these class extensions to populate CIs and discover various technologies and software.

Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

Unmatched APIs

API endpoints that are not structured well enough to populate the API [cmdb_ci_api] and API Component [cmdb_ci_api_component] classes instead populate the Unmatched API Endpoint [cmdb_ci_unmatched_api_endpoint] class. Unmatched APIs are typically used by integrations like security scanners or observability tools.

Classes

This section lists the classes that the CMDB CI Class Models store app adds or updates.

CMDB CI Class Models: Release 1.49.0 adds the following class for unmatched API endpoints. For the list of CMDB classes in a base system, including ones that this store app might be extending, see [CMDB tables descriptions](#).

Class	Extends	Description
Unmatched API Endpoint [cmdb_ci_unmatched_api_endpoint]	Configuration Item [cmdb_ci]	APIs with unstructured endpoints that cannot populate cmdb_ci_api or cmdb_ci_api_component.

Class attributes

CMDB CI Class Models: Release 1.49.0 adds the following attributes to the Unmatched API Endpoint [cmdb_ci_unmatched_api_endpoint] class.

Unmatched API Endpoint [cmdb_ci_unmatched_api_endpoint]

Attribute	Data type	Description
Method	String	Backend protocol types of the API. Examples: <ul style="list-style-type: none">• Lambda• HTTP• Logic App
Protocol	String	Communication protocol. Example: HTTP, HTTPS.
URL	String (1024)	URL of the resource being called.
ID	String (1024)	Unique identifier from the source system.
Internet Facing	Boolean	Boolean that denotes whether the component is reachable from the public internet. Specify 1 or "true" if the component is reachable.
Authorization	String	Type of authorization or authentication method. Example:

Attribute	Data type	Description
		<ul style="list-style-type: none"> • Basic • Key • Oauth • None
Request data types	String (255)	List of data types in the request. Examples: <ul style="list-style-type: none"> • CC • Email • Address
Response data types	String (255)	List of data types in the response. Examples: <ul style="list-style-type: none"> • CC • Email • Address

Key Relationship Structures

The Unmatched API Endpoint [cmdb_ci_unmatched_api_endpoint] CMDB class and API Endpoint Discovered [api_endpoint_discovered] non-CMDB table are intended for scenarios where unstructured API data is ingested and needs to be processed beyond the capabilities of IntegrationHub-ETL.

API Endpoint Discovered [api_endpoint_discovered] can serve as a staging table for raw data for the API [cmdb_ci_api] and API Component [cmdb_ci_api_component] classes. API data that does not come from a well structured data source or cannot have standard identification rules applied can first populate the API Endpoint

Discovered [api_endpoint_discovered] table. You can then use a customer-defined integration to ensure that well-formed data that matches the quality and condition requirements for API [cmdb_ci_api] and API Component [cmdb_ci_api_component] populates the most appropriate table.

In cases where the data cannot be correctly parsed or is structured in an unexpected fashion, you can instead use a customer-defined integration to populate the Unmatched API Endpoint [cmdb_ci_unmatched_api_endpoint] table.

Data that populates the API Endpoint Discovered [api_endpoint_discovered] table is purged every 30 days.

API Endpoint Discovered [api_endpoint_discovered]

Attribute	Data type	Description
Name	String (100)	Name of the API endpoint.
Method	String	REST API methods. Examples: <ul style="list-style-type: none">• GET• POST• DELETE
URL	String (1024)	URL of the resource being called.
Type	Choice list	Type of API. You can specify: <ul style="list-style-type: none">• REST• SOAP• HTTP• gRPC

Attribute	Data type	Description
		<ul style="list-style-type: none"> • GraphQL • Websocket
Authorization	String	<p>Type of authorization or authentication method. Example:</p> <ul style="list-style-type: none"> • Basic • Key • Oauth • None
Internet Facing	Boolean	Boolean that denotes whether the component is reachable from the public internet. Specify 1 or "true" if the component is reachable.
Request Datatypes	String (255)	<p>List of data types in the request. Examples:</p> <ul style="list-style-type: none"> • CC • Email • Address
Response Datatypes	String (255)	<p>List of data types in the response. Examples:</p> <ul style="list-style-type: none"> • CC

Attribute	Data type	Description
		<ul style="list-style-type: none"> Email Address
Headers	String (255)	Comma-separated list of header names.
Configuration Item	Reference	Reference to a matching CMDB CI (API Component or Unmatched API Endpoint).

Related non-CMDB tables

The Unmatched API extension class uses the API Deployment [api_deployment] non-CMDB table as a related list:

API Deployment [api_deployment]

Attribute	Data type	Description
Name	String (100)	Name of the API deployment.
API	Reference	Reference to the deployed API (cmdb_ci_api).
Unmatched API Endpoint	Reference	Reference to the unmatched API endpoint, if the API doesn't match an existing API (cmdb_ci_unmatched_api_endpoint)
Configuration Item	Reference	Reference to the Configuration Item. This is typically

Attribute	Data type	Description
		manually specified as a reference, if you know what CI the API is deployed to.

Avi load balancer extension classes

The CMDB CI Class Models store app adds or updates classes for the Avi load balancer.

The app adds class models that extend the CMDB class hierarchy, including class descriptions, identification rules, identifier entries, and dependent relationships if applicable. You can use the added classes as any other CMDB class. Applications such as Discovery and Service Mapping patterns can use these class extensions to populate CIs and discover various technologies and software.

Request apps on the Store

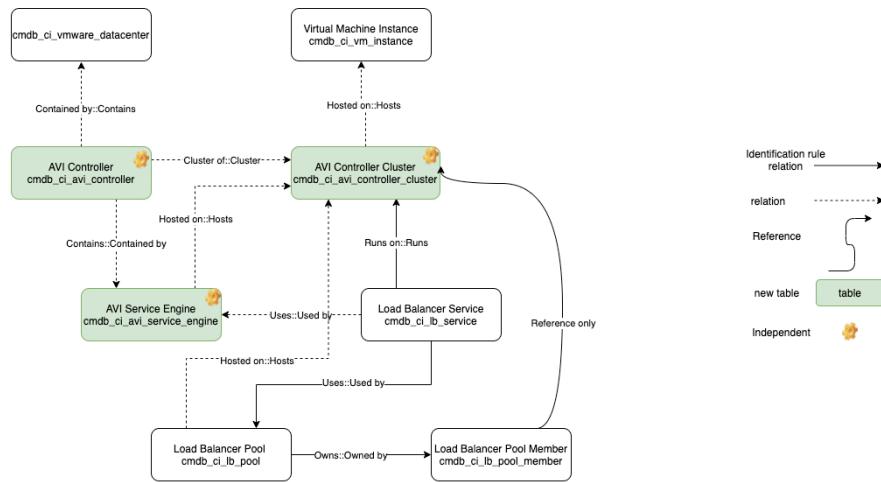
Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

Avi load balancer

The Avi Vantage platform is built on software-defined principles, enabling a next generation architecture to deliver the flexibility and simplicity expected by IT and lines of business. The Avi Vantage platform architecture separates the data and control planes to deliver application services beyond load balancing, such as application analytics, predictive autoscaling, micro-segmentation, and self-service, for app owners in on-premises or cloud environments. The platform provides a centrally managed, dynamic pool of load balancing resources on commodity x86 servers, virtual machines, or containers, to deliver granular services close to individual applications. Providing these services allows network services to scale near infinitely without the added complexity of managing hundreds of disparate appliances.

ServiceNow Discovery uses the [Avi Vantage load balancer discovery](#) pattern to find Avi load balancer resources.

Avi load balancer classes integrated with the CMDB class hierarchy



Classes

This section lists the classes that the CMDB CI Class Models store app adds or updates. CMDB CI Class Models: Release 1.6.0 adds the following classes for the Avi load balancer. For the list of classes in a base system, including classes that this store app might be extending, see [CMDB tables descriptions](#).

Class	Extends	Description
Avi Controller [cmdb_ci_avi_controller]	Virtual Machine Object [cmdb_ci_vm_object]	Avi Controller is a single point of management and control that is the 'brain' of the entire Avi Vantage system, and typically deployed as a redundant three-node cluster.
Avi Controller Cluster	Virtual Machine Object	Avi Controller cluster uses big data analytics to analyze the

Class	Extends	Description
[cmdb_ci_avi_controller_cluster]	[cmdb_ci_vm_object]	data and present actionable insights to administrators on intuitive dashboards on the Avi Admin Console.
Avi Service Engine [cmdb_ci_avi_service_engine]	Virtual Machine Object [cmdb_ci_vm_object]	Avi Service Engines (SEs) handle all data plane operations within Avi Vantage by receiving and executing instructions from the Avi Controller.

Class columns

CMDB CI Class Models: Release 1.6.0 adds the following column to the respective class.

Avi Service Engine [cmdb_ci_avi_service_engine]

Added column	Description
version	The version of the Avi Service Engine resource.

Related concepts

- [CMDB schema model](#)

BYOL model of RDS for Oracle extension classes

The CMDB CI Class Models store app adds or updates classes for the BYOL Model of RDS for Oracle.

The app adds class models that extend the CMDB class hierarchy, including class descriptions, identification rules, identifier entries, and dependent relationships if applicable. You can use the added classes as any other CMDB class. Applications such as Discovery and Service

Mapping patterns can use these class extensions to populate CIs and discover various technologies and software.

Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

BYOL Model of RDS for Oracle

The Amazon RDS for Oracle is a fully managed commercial database that makes it easy to set up, operate, and scale Oracle deployments in the cloud. You can run Amazon RDS for Oracle under two different licensing models – “License Included” and “Bring-Your-Own-License (BYOL)”. In the “License Included” service model, you do not need separately purchased Oracle licenses; the Oracle Database software has been licensed by AWS.

We support discovery of the RDS Databases and their licenses.

Classes

This section lists the classes that the CMDB CI Class Models store app adds or updates. CMDB CI Class Models: Release 1.23.0 adds the following classes for the BYOL Model of RDS for Oracle. For the list of classes in a base system, including classes that this store app might be extending, see [CMDB tables descriptions](#).

Class	Description
cmdb_ci_cloud_database	The cloud databases.
cmdb_ci_serverless_hardware	Hardware type information of the databases.

Class Columns

CMDB CI Class Models: Release 1.23.0 adds the following column to the respective class.

Serverless Hardware [cmdb_ci_serverless_hardware]

Added column	Description
cloud_vendor	The cloud vendor.
host_type	The host type such as PaaS/IaaS.
cpu_core_count	Amount of CPU cores.
cpu_core_thread	Amount of CPU threads.
cpu_count	Amount of CPUs.
object_id	ID of the CI.

Cloud Database [cmdb_ci_cloud_database]

Added column	Description
multi_az	Determines if the database is deployed on multiple availability zones (true/false).
replication_enabled	Determines if replication is enabled (true/false).
replication_type	Replication type.
replica_source	Database name of the replication source.

Firewall extension classes

The CMDB CI Class Models store app adds or updates classes for firewall devices.

The app adds class models that extend the CMDB class hierarchy, including class descriptions, identification rules, identifier entries, and dependent relationships (if applicable). You can use the added classes as any other CMDB class. Applications such as Discovery and Service

Mapping patterns can use these class extensions to populate Cls and discover various technologies and software.

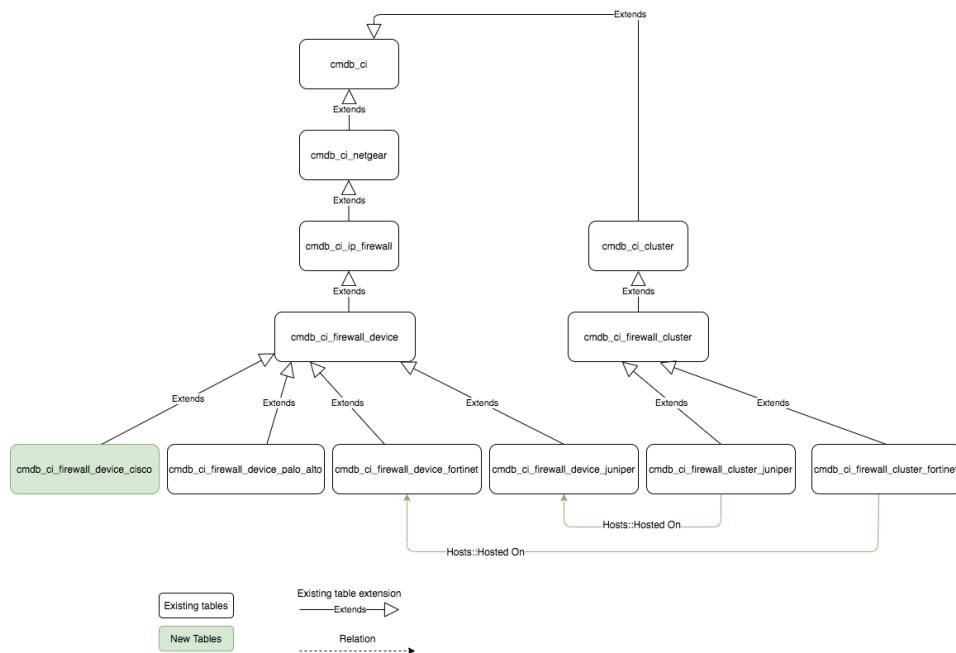
Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

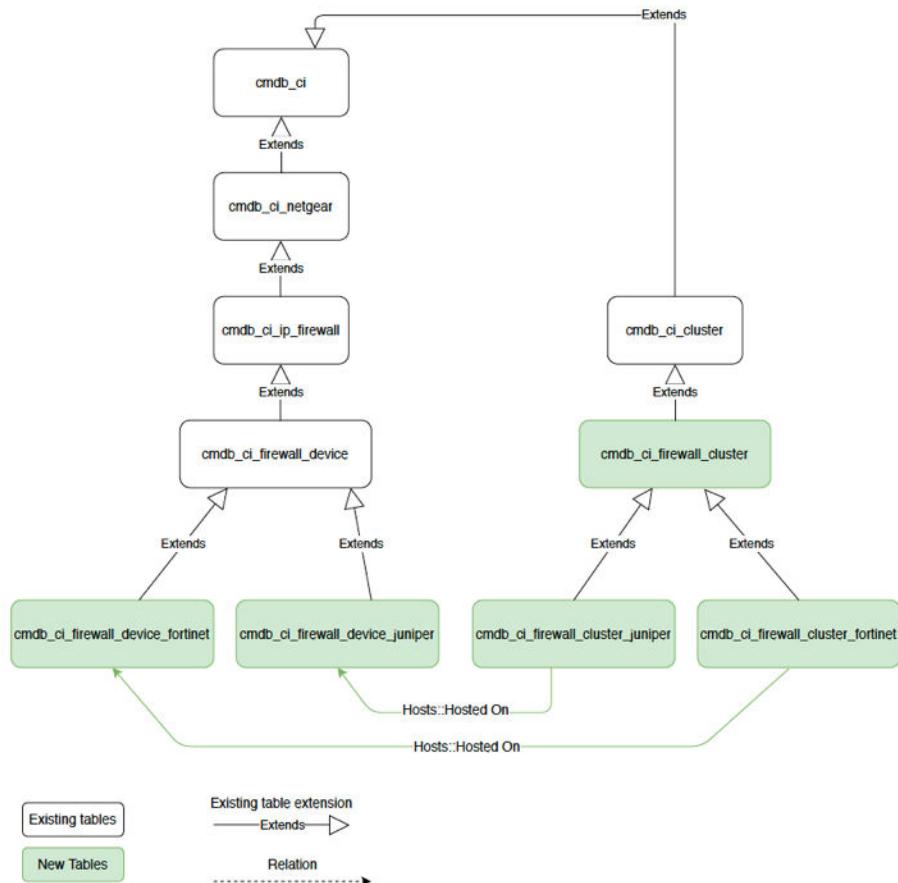
Firewalls

A firewall is a network security system that monitors and controls incoming and outgoing network traffic, based on security policies. Firewalls typically form a barrier between an internal network and an untrusted external network, such as the internet. It usually consists of security policies that help secure an organization from external threats and cyber attacks. Firewall vendors may provide a centralized firewall manager to manage many firewall devices and the security policies residing on them. For example, Panorama™ is the centralized management system for Palo Alto Networks firewalls.

Firewall extension classes integrated with the CMDB class hierarchy
(CMDB CI Class Models: Release 1.11.0)



Firewall extension classes integrated with the CMDB class hierarchy
(CMDB CI Class Models: Release 1.10.0)



Classes

This section lists the classes that the CMDB CI Class Models store app adds or updates.

CMDB CI Class Models: Release 1.10.0 adds or updates the following classes for the discovery of network firewall devices. For the list of CMDB classes in a base system, including ones that this store app might be extending, see [CMDB tables descriptions](#).

Class	Extends	Description
IP Firewall [cmdb_ci_ip_firewall]	NETGEAR [cmdb_ci_netgear]	Contains all network firewalls.
Firewall Device [cmdb_ci_firewall_device]	IP Firewall [cmdb_ci_ip_firewall]	Network security system that monitors and controls incoming and outgoing network traffic, based on security policies.
Fortinet Firewall Device [cmdb_ci_firewall_device_fortinet]	Firewall Device [cmdb_ci_firewall_device]	Fortinet firewall device.
Juniper Firewall Device [cmdb_ci_firewall_device_juniper]	Firewall Device [cmdb_ci_firewall_device]	Juniper firewall device.
Firewall Device Group [cmdb_ci_firewall_device_group]	CMDB CI [cmdb_ci]	Group of firewall devices.
Panorama Firewall Device Group [cmdb_ci_firewall_device_group_panorama]	Firewall Device Group [cmdb_ci_firewall_device_group]	Group of Panorama firewall devices.
Palo Alto Firewall Device [cmdb_ci_firewall_device_palo_alto]	Firewall Device [cmdb_ci_firewall_device]	Palo Alto firewall device.
Firewall Cluster [cmdb_ci_firewall_cluster]	[cmdb_ci_cluster]	Group of firewall nodes that work as a single logical entity.
Fortinet Firewall Cluster [cmdb_ci_firewall_cluster_fortinet]	Firewall Cluster [cmdb_ci_firewall_cluster]	Fortinet firewall cluster.

Class	Extends	Description
Juniper Firewall Cluster [cmdb_ci_firewall_cluster_juniper]	Firewall Cluster [cmdb_ci_firewall_cluster]	Juniper firewall cluster.
Firewall Manager [cmdb_ci_firewall_manager]	CMDB CI [cmdb_ci]	System that provides centralized management for many firewall devices and the security policies residing on them.
Panorama Firewall Manager [cmdb_ci_firewall_manager_panorama]	Firewall Manager [cmdb_ci_firewall_manager]	The centralized network security management tool for Palo Alto Networks firewalls.
Firewall Security Policy [cmdb_ci_firewall_sec_policy]	CMDB CI [cmdb_ci]	The security policy that the firewall device enforces.
Panorama Firewall Security Policy [cmdb_ci_firewall_sec_policy_panorama]	Firewall Security Policy [cmdb_ci_firewall_sec_policy]	The security policy that the Panorama firewall device enforces.

CMDB CI Class Models: Release 1.12.0 adds the following class for the discovery of network firewall devices.

Class	Extends	Description
Cisco Firewall Device [cmdb_ci_firewall_device_cisco]	Firewall Device [cmdb_ci_firewall_device]	All Cisco Firewall devices.

Class columns

CMDB CI Class Models: Release 1.10.0 adds the following columns to the respective classes.

IP Firewall [cmdb_ci_ip_firewall] class

Added columns	Description
Hardware Operating System	OS running on the hardware.
Hardware OS Version	OS version running on the hardware.

Firewall Cluster [cmdb_ci_firewall_cluster] class

Added columns	Description
Hardware Operating System	OS running on the hardware.
Hardware OS Version	OS version running on the hardware.

CMDB CI Class Models: Release 1.12.0 adds no columns to the existing classes.

Related concepts

- [CMDB schema model](#)

IBM Hardware Management Console (HMC) extension classes

The CMDB CI Class Models store app adds or updates classes for the IBM Hardware Management Console (HMC).

The app adds class models that extend the CMDB class hierarchy, including class descriptions, identification rules, identifier entries, and dependent relationships if applicable. You can use the added classes as any other CMDB class. Applications such as Discovery and Service Mapping patterns can use these class extensions to populate CIs and discover various technologies and software.

Request apps on the Store

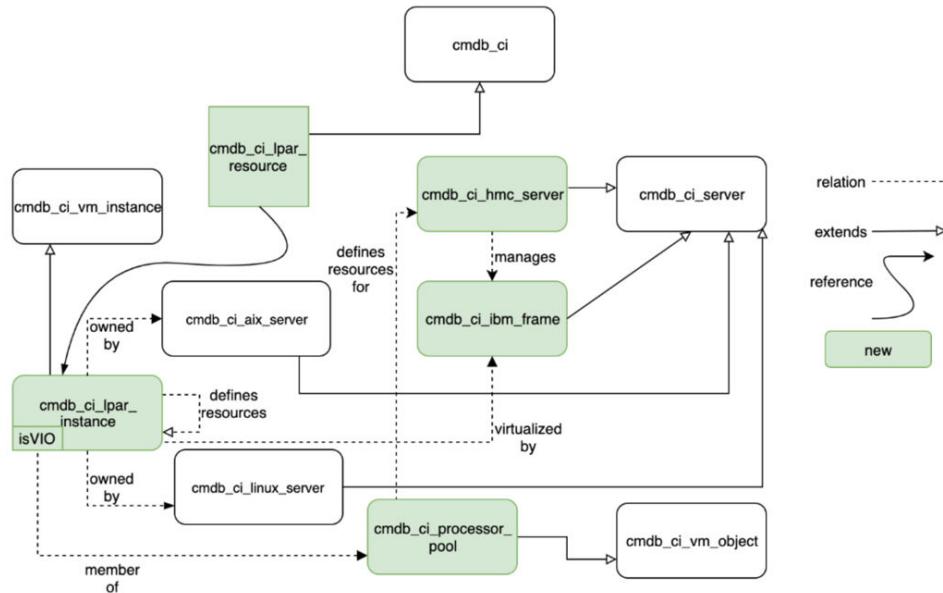
Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

IBM Hardware Management Console (HMC)

The IBM HMC extension classes support discovery of IBM virtualization technology by providing:

- Discovery of LPARs/FRAAMES, which SAM use cases need
- Topology data of IBM HMC, which event correlation requires
- Topology data for ITSM use cases, such as in-frame migration

IBM HMC extension classes integrated with the CMDB class hierarchy



Classes

This section lists the classes that the CMDB CI Class Models store app adds or updates. See the class columns table for further details about the columns added for each class.

CMDB CI Class Models: Release 1.3.0 adds the following classes for the IBM HMC. For the list of CMDB classes in a base system, including ones that this store app might be extending, see [CMDB tables descriptions](#).

Class	Extends	Description
IBM Frame [cmdb_ci_ibm_frame]	Server [cmdb_ci_server]	IBM physical machine with considerable resources which can be virtualized.
IBM HMC Processor pool [cmdb_ci_processor_pool]	Virtual Machine Object [cmdb_ci_vm_object]	IBM shared pool used to allocate processors to a group of LPARs.

CMDB CI Class Models: Release 1.2.0 adds the following classes for the IBM HMC.

Class	Extends	Description
IBM HMC Server [cmdb_ci_hmc_server]	Server [cmdb_ci_server]	IBM console that manages frames and assigns logical partitions (LPARs) to pools.
IBM Frame [cmdb_ci_ibm_frame]	Server [cmdb_ci_server]	IBM physical machine with considerable resources which can be virtualized.
IBM LPAR Instance [cmdb_ci_lpar_instance]	Virtual Machine Instance	IBM logical partition representing the virtual

Class	Extends	Description
	[cmdb_ci_vm_instance]	aspect of the operating system.
LPAR Resource [cmdb_ci_lpar_resource]	Configuration Item [cmdb_ci]	Resource of an LPAR instance.
IBM HMC Processor pool [cmdb_ci_processor_pool]	Virtual Machine Object [cmdb_ci_vm_object]	IBM shared pool used to allocate processors to a group of LPARs.

Class columns

CMDB CI Class Models: Release 1.3.0 adds the following columns to the respective classes.

IBM Frame [cmdb_ci_ibm_frame] class

Added columns	Description
Current available processor units	Current available processor units.
Configurable processor units	Configurable processor units.
Configurable memory units	Configurable memory units.
Installed processor units	Installed processor units.
Current available memory units	Current available memory units.
Installed memory units	Installed memory units.

IBM HMC Processor pool [cmdb_ci_processor_pool] class

Added columns	Description
LPAR IDs	LPAR IDs.
LPAR names	LPAR names.

Added columns	Description
Frame name	Frame name.

CMDB CI Class Models: Release 1.2.0 adds the following columns to the respective classes.

IBM HMC Server [cmdb_ci_hmc_server] class

Added columns	Description
Frame count	Count of frames.

Added columns	Description
Is VIO	Flags whether this VM is a (VIO) virtual input/output server in the HMC topology.
High Watermark VCPU	Peak in the utilization of virtual CPU assigned to during the reporting period.
Frame Serial Number	Serial number of a frame (frame being a physical machine such as ESX).
VIO Servers	List of VIO servers.

LPAR Resource [cmdb_ci_lpar_resource] class

Added columns	Description
Node name	Name of the node.
Partition Name	Name of the partition.
Partition Number	Number of the partition.
Type	Type of LPAR Resource.

Added columns	Description
Mode	Mode of LPAR Resource.
Entitled Capacity	Allotted capacity granted.
Partition Group-ID	ID of a group of partitions.
Shared Pool ID	ID of a pool of shared processors.
Online Virtual CPUs	A virtual CPU.
Maximum Memory	Maximum amount of memory.
Minimum Memory	Minimum amount of memory.
Variable Capacity Weight	Logical partition processor capacity weight.
Minimum Capacity	Minimum number of processes.
Capacity Increment	Increments of process.
Maximum Physical CPUs in system	Maximum CPUs allotted in system.
Active Physical CPUs in system	Current CPUs in system.
Active CPUs in Pool	Number of active CPUs within a pool.
Shared Physical CPUs in system	Number of shared CPUs within a system.
Maximum Capacity of Pool	Maximum capacity of processes within a pool.
Entitled Capacity of Pool	Number of processes that are entitled.
Unallocated Capacity	Number of free spaces.

Added columns	Description
Physical CPU Percentage	Number of CPUs allocated to system.
Unallocated Weight	At no extra charge resources on instance.
Desired Virtual CPUs	Target number of virtual CPUs.
Desired Memory	Target amount of memory.
Desired Variable Capacity Weight	Targeted processor load.
Desired Capacity	Target resources used within instance.
High Watermark VCPU	Peak in the utilization of virtual CPU assigned to during the reporting period.

IBM HMC Processor pool [cmdb_ci_processor_pool] class

Added columns	Description
Pool ID	ID of pool of processors.
CPU Core count	Number of CPU cores.
Memory count	Amount of memory used.
Unassigned cores	Number of unused cores.
Unassigned memory	Amount of unassigned memory.

The following class has no added columns: IBM Frame [cmdb_ci_ibm_frame].

Related concepts

- CMDB schema model

Internet of Things (IoT) extension classes

The CMDB CI Class Models store app adds or updates classes for the Internet of Things (IoT).

The app adds class models that extend the CMDB class hierarchy, including class descriptions, identification rules, identifier entries, and dependent relationships if applicable. You can use the added classes as any other CMDB class. Applications such as Discovery and Service Mapping patterns can use these class extensions to populate CIs and discover various technologies and software.

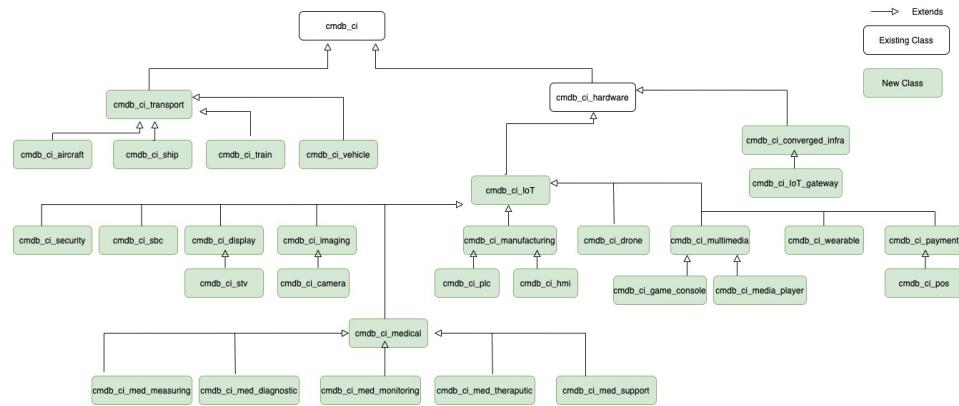
Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

Internet of Things (IoT)

IoT is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. The classes added in this release, extend the Data Model to provide a foundation for the representation of IoT CI classes. This foundation underpins workflows for Enterprise Asset Management (EAM), Governance Risk Compliance (GRC), Component Supply Management (CSM), and Field Service Management (FSM) surrounding the management of IoT devices and the transport vehicles that some reside in.

IoT extension classes integrated with the CMDB class hierarchy



Classes

This section lists the classes that the CMDB CI Class Models store app adds or updates.

CMDB CI Class Models: Release 1.6.0 adds the following classes for IoT. For the list of CMDB classes in a base system, including ones that this store app might be extending, see [CMDB tables descriptions](#).

Class	Extends	Description
Transport Type [cmdb_ci_transport]	<code>cmdb_ci</code>	Types of transportation that contain interconnected technology.
Converged Infrastructure [cmdb_ci_converged_infra]	<code>cmdb_ci_hardware</code>	Devices that serve both computing and networking functions.
IoT Device [cmdb_ci_IoT]	<code>cmdb_ci_hardware</code>	Parent table that contains Internet of Things device types.

Class	Extends	Description
Aircraft [cmdb_ci_aircraft]	Transport Type [cmdb_ci_transport]	A transportation method that utilizes air or space as its primary pathway. For example, airplanes and helicopters.
Ship [cmdb_ci_ship]	Transport Type [cmdb_ci_transport]	A transportation method that utilizes water as its primary pathway. For example, ships.
Train [cmdb_ci_train]	Transport Type [cmdb_ci_transport]	A transportation method that utilizes rails as its primary pathway. For example, Amtrak.
Vehicle [cmdb_ci_vehicle]	Transport Type [cmdb_ci_transport]	A transportation method that utilizes wheels or tracks as its method of movement. For example, cars, trucks, bulldozers.
IoT Gateway [cmdb_ci_IoT_gateway]	Converged Infrastructure [cmdb_ci_converged_infra]	<p>A device that provides the following services:</p> <ul style="list-style-type: none"> Forwards packets between LAN and WAN on the IP layer. Performs application layer functions between IoT nodes and other entities. Enables local, short-range communication

Class	Extends	Description
		between IoT devices.
Security Device [cmdb_ci_security]	IoT Device [cmdb_ci_IoT]	Connected device that serves a security function such as badge readers.
Single Board Computing [cmdb_ci_sbc]	IoT Device [cmdb_ci_IoT]	Single Board Computing device such as a Raspberry Pi.
Display Device [cmdb_ci_display]	IoT Device [cmdb_ci_IoT]	Connected device that displays images.
Imaging Device [cmdb_ci_imaging]	IoT Device [cmdb_ci_IoT]	Connected device that captures images.
Medical Device [cmdb_ci_medical]	IoT Device [cmdb_ci_IoT]	Connected device that serves a Medical Care function such as a nurse call unit.
Manufacturing Device [cmdb_ci_manufacturing]	IoT Device [cmdb_ci_IoT]	Connected device that helps a manufacturing process.
Multimedia Device [cmdb_ci_multimedia]	IoT Device [cmdb_ci_IoT]	A connected device that helps the generation or delivery of media content.

Class	Extends	Description
Payment Device [cmdb_ci_payment]	IoT Device [cmdb_ci_IoT]	Connected device that allows for purchasing goods or services.
Drone [cmdb_ci_drone]	IoT Device [cmdb_ci_IoT]	Unmanned connected device with mobility.
Wearable Technology [cmdb_ci_wearable]	IoT Device [cmdb_ci_IoT]	Connected device that is worn by an entity such as a smart watch.
Smart Television [cmdb_ci_stv]	Display Device [cmdb_ci_display]	A television that is network connected and can run applications.
IP Camera [cmdb_ci_ip_camera]	Imaging Device [cmdb_ci_imaging]	A camera that is network connected.
Medical Measuring Device [cmdb_ci_med_measuring]	Medical Device [cmdb_ci_medical]	A device that is network connected which takes medical measurements such as the Abbott iStat System.
Medical Diagnostic Device [cmdb_ci_med_diagnostic]	Medical Device [cmdb_ci_medical]	A medical diagnostic device that is network connected, such as a CT system.
Medical Monitoring Device	Medical Device	A device that is network connected

Class	Extends	Description
[cmdb_ci_med_monitoring]	[cmdb_ci_medical]	which is used to monitor medical patients. For example, the Phillips Patient Monitoring System.
Medical Therapeutic Device [cmdb_ci_med_therapeutic]	Medical Device [cmdb_ci_medical]	A device that is network connected which provides medical patient therapy. For example, the Compex Muscle Stimulator.
Medical Support Device [cmdb_ci_med_support]	Medical Device [cmdb_ci_medical]	A device that is network connected that helps deliver medical care. For example, the Pyxis MedStation.
Process Logic Controller [cmdb_ci_plc]	Manufacturing Device [cmdb_ci_manufacturing]	A logic controller that is network connected which is used in manufacturing. For example, devices made by Siemens and Allen Bradley.
Human Machine Interface [cmdb_ci_hmi]	Manufacturing Device [cmdb_ci_manufacturing]	An HMI that is network connected which is used in manufacturing. For example, devices made by Siemens and Allen Bradley.
Game Console	Multimedia Device [cmdb_ci_multimedia]	A device that is network connected which is used to play games or stream

Class	Extends	Description
[cmdb_ci_game_console]		media. For example, an Xbox or Playstation.
Media Player [cmdb_ci_media_player]	Multimedia Device [cmdb_ci_multimedia]	A device that is network connected which is used to play digital media content. For example, Amazon Fire TV.
Display Monitor Control [cmdb_ci_monitor_control]	Multimedia Device [cmdb_ci_multimedia]	A device that is network connected that controls the display of media on a monitor. For example, a Crestron Media Controller.
Point of Sale Device [cmdb_ci_pos]	Payment Device [cmdb_ci_payment]	A device that is network connected which is used in the purchase of goods or services. For example, a credit card reader.

Related concepts

- [CMDB schema model](#)

Kong extension classes

The CMDB CI Class Models store app adds or updates classes for Kong gateways.

The app adds class models that extend the CMDB class hierarchy, including class descriptions, identification rules, identifier entries, and dependent relationships (if applicable). You can use the added classes as any other CMDB class. Applications such as Discovery and Service

Mapping patterns can use these class extensions to populate CIs and discover various technologies and software.

Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

Kong

Kong is an API management platform that enables enterprise companies to better manage client and host traffic.

Classes

This section lists the classes that the CMDB CI Class Models store app adds or updates.

CMDB CI Class Models: Release 1.49.0 adds the following classes for Kong. For the list of CMDB classes in a base system, including ones that this store app might be extending, see [CMDB tables descriptions](#).

Class	Extends	Description
Kong Gateway [cmdb_ci_kong_gateway]	API Gateway [cmdb_ci_api_gateway]	The Kong gateway application that hosts and manages individual APIs. Example: Kong Gateway instanceName.
Kong Load Balancer [cmdb_ci_kong_lb]	Load Balancer Application [cmdb_ci_lb_appl]	The out-of-box load balancer on the Kong gateway application that points to backend service instances when fulfilling API requests. Example: httpbin-upstream.

Class	Extends	Description
Kong Target [cmdb_ci_kong_target]	API Component [cmdb_ci_api_component]	The load balanced backend of the gateway that fulfills API requests. Example: httpbin-target1.

Class attributes

CMDB CI Class Models: Release 1.49.0 adds the following attributes to the respective classes.

Kong Gateway [cmdb_ci_kong_gateway]

Attribute	Data type	Description
Admin URL	String (255)	URL for making admin API requests.
Database	String	Type of database used by the Kong gateway. Example: Postgres or Cassandra.

Kong Load Balancer [cmdb_ci_lb_appl]

Attribute	Data type	Description
Algorithm	String	Type of algorithm used for load balancing. Example: round robin.
ID	String (255)	Unique identifier from the source system.

Kong Target [cmdb_ci_kong_target]

Attribute	Data type	Description
Target	String (255)	URL of target integration.

Key Relationship Structures

There are a number of key relationships that need to be defined for API and Kong classes.

Kong relationships

Parent class	Relationship	Child class	Relationship type
API Backend [cmdb_ci_api_b ackend]	Uses::Used By	Kong Load Balancer	Suggested
Kong Load Balancer [cmdb_ci_lb_ap pl]	Contains::Contai ned By	Kong Target	Dependent
Kong Gateway [cmdb_ci_kong_ gateway]	Provides::Provid ed By	Kong Load Balancer	Dependent

Related non-CMDB tables

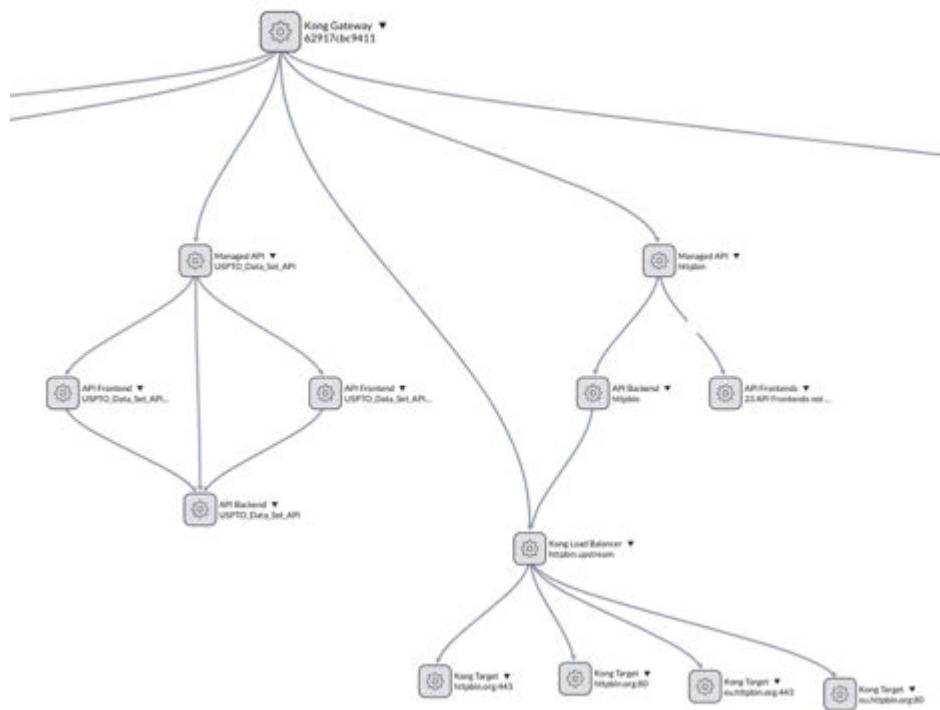
The Kong Gateway class uses the Kong Workspace non-CMDB table as a related list:

Kong Workspace [kong_workspace]

Attribute	Data type	Description
Name	String (100)	Name of the Kong workspace.
ID	String (255)	Unique identifier from the source system.
API Gateway	Reference	Reference to the Kong API gateway.

Kong gateway example

Here is an example of a dependency view for the Kong gateway class that shows how a gateway would populate the dependent managed API dependent class with related APIs and components. The Managed API class is considered a first level relationship with respect to the gateway, while the frontend and backend components are considered second level relationships. From here, you can then bind alerts to these Cls, configure dynamic Cls for service views and incidents, or establish any additional workflows that use Cls.



Kubernetes extension classes

The CMDB CI Class Models store app adds or updates classes for the Kubernetes pattern.

The app adds class models that extend the CMDB class hierarchy, including class descriptions, identification rules, identifier entries, and dependent relationships if applicable. You can use the added classes as any other CMDB class. Applications such as Discovery and Service Mapping patterns can use these class extensions to populate Cls and discover various technologies and software.

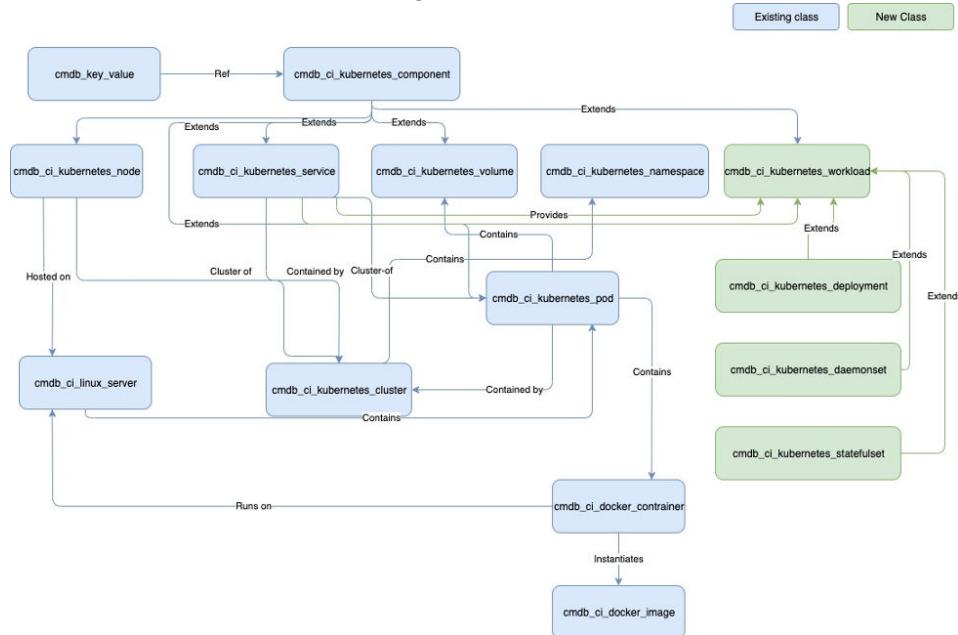
Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

Kubernetes pattern

The Kubernetes pattern main flow helps with discovering Kubernetes core elements. The classes in this release, extend the support to discover kubernetes workload controller components like deployments, daemonsets, and statefulsets. The Workload Share library captures information about deployments, daemonsets, and statefulsets and stores them in the respective tables. Other extensions include a YAML and service mesh extension that generates a YAML file to track configuration files and creating service to service relations by discovering service mesh information.

Kubernetes extension class integrated with the CMDB hierarchy



Kubernetes workload

This screenshot shows the details of the `Kubernetes Workload` class in the CMDB CI Class Models store app.

- Header:** Hierarchy, Configuration Item > Kubernetes Component > Kubernetes Workload
- Class Info:**
 - Basic Info
 - Attributes
 - Identification Rule
 - Dependent Relationships
 - Reconciliation Rules
 - Suggested Relationships
 - All Relationship Rules (selected)
- Relationship Rules:**
 - Provides**: A relationship rule connecting `Kubernetes Service` to `Kubernetes Workload`.
 - Hosted on**: A relationship rule connecting `Kubernetes Cluster` to `Kubernetes Workload`.
 - Contains**: A relationship rule connecting `Tracked Configuration...` to `Kubernetes Workload`.

Classes

This section lists the classes that the CMDB CI Class Models store app adds or updates.

CMDB CI Class Models: Release 1.12.0 adds the following classes for Kubernetes pattern. For the list of CMDB classes in a base system, including ones that this store app might be extending, see [CMDB tables descriptions](#).

Class	Extends	Fields	Relation
cmdb_ci_kubernetes_workload	cmdb_ci_kubernetes_components		Provides from cmdb_ci_kubernetes_service
cmdb_ci_kubernetes_deployment	cmdb_ci_kubernetes_workload	<ul style="list-style-type: none"> • Replicas Desired • Replicas Updated • Replicas Total • Replicas Available • Replicas Unavailable 	Hosted on Cluster
cmdb_ci_kubernetes_daemonset	cmdb_ci_kubernetes_workload	<ul style="list-style-type: none"> • Pods running • Pods Waiting • Pods Succeeded • Pods Failed • Pods Available 	Hosted on Cluster
cmdb_ci_kubernetes_statefulset	cmdb_ci_kubernetes_workload	<ul style="list-style-type: none"> • Pods running • Pods Waiting 	Hosted on Cluster

Class	Extends	Fields	Relation
		<ul style="list-style-type: none">• Pods Succeeded• Pods Failed• Pods Available	

Related concepts

- [CMDB schema model](#)

Network Intrusion Detection System (NIDS) CI extension class

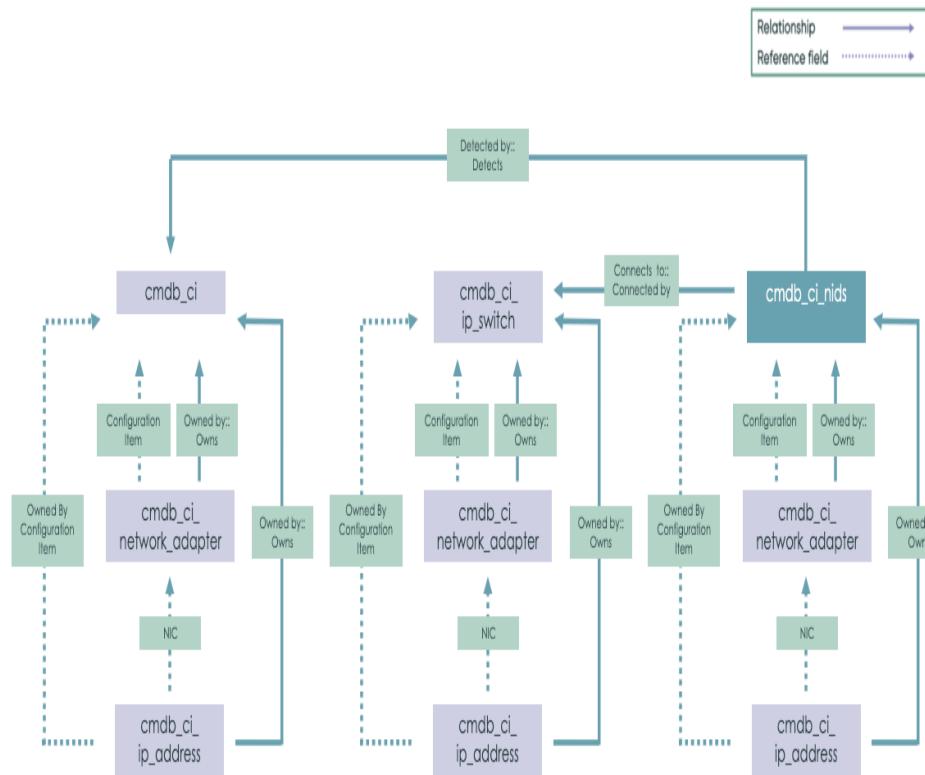
The Network Intrusion Detection System (NIDS) [cmdb_ci_nids] class builds the relationships between passive network monitoring appliances, and the devices on the network that it discovers. A NIDS Manager manages the NIDS sensors that detect the devices and builds "Detects::detected by" relationships between the NIDS records (parent) and the CIs it discovers (child).

This topic lists the relevant classes that the CMDB CI Class Models store app adds or updates. See the class columns table for further details about the columns added for each class.

Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

Network Intrusion Detection System (NIDS) schema structure



Classes

CMDB CI Class Models: Release 1.30 adds the following classes for the Network Intrusion Detection System (NIDS). For the list of CMDB classes in a base system, including ones that this store app might be extending, see [CMDB tables descriptions](#).

Class	Extends	Description
Network Intrusion Detection System (NIDS) (cmdb_ci_nids)	cmdb_ci_ids_network	NIDS is an intrusion detection system within the network that examines the traffic from all devices on the network.

Class	Extends	Description
		NIDS scanners build relationships between the OT network scanning appliances, and the OT assets on the network. An NIDS Manager manages the NIDS sensors.

Class columns

CMDB CI Class Models: Release 1.30 adds the following columns to the Network Intrusion Detection System (NIDS) [cmdb_ci_nids] class.

Network Intrusion Detection System (NIDS) [cmdb_ci_nids] class

Column label	Column name	Description
NIDS source ID	Correlation_id	Identifier of the NIDS asset. Uses the assigned Correlation ID for the NIDS as its nids_source_id.
NIDS source name	nids_source_name	Name of the NIDS asset.
Network type assignment	network_type_assignment	Designates if the asset is on an IT or OT network.
NIDS assignment site	isa_entity_site	ISA site assigned to the NIDS. This information is available when the logged in user has an assigned ISA Admin role.

Roles and Access Control Logic (ACLs)

The NIDS Admin (cmdb_nids_admin) role is associated with the Network Intrusion Detection System (NIDS) [cmdb_ci_nids] class: Can create, read, update, and delete Network IDS (NDIS) OT records. To view the Network IDS Application selection on the application menu, you must have this role.

Key relationship structure

For each CMDB CI record with a “Detected by” relationship with an NIDS record, a ServiceNow Operational Technology Certified Service Graph Connector does the following:

1. Assigns the following NIDS-related metadata values to the CI:
 - a. Location
 - b. Company
 - c. Related users (Owned by, Managed by, Supported by, Assigned to)
 - d. Related user groups (Approval group, Managed by Group, Support group, Change group)
2. If the NIDS network type is set to OT, it assigns the following NIDS-related metadata values to the CI:
 - a. Creates an OT Asset (cmdb_ot_entity) record for the CI, using the cmdb_ot_entity reference on the CI.
 - b. Assigns the NIDS assignment zone to the OT Asset record.
 - c. If the Industrial Process Manager is installed, assign the NIDS assignment site to the OT Asset record.

Network Inventory (NI) extension classes

The CMDB CI Class Models store app adds or updates classes for the Telecommunications Network Inventory application. Telecommunications Network Inventory uses the Network Inventory (NI) extension classes to extend the Configuration Management Database (CMDB) Configuration Item (CI) class hierarchy.

These extensions enable the CMDB to store information about a service provider's network inventory. The store app adds class models that extend the CMDB class hierarchy, including:

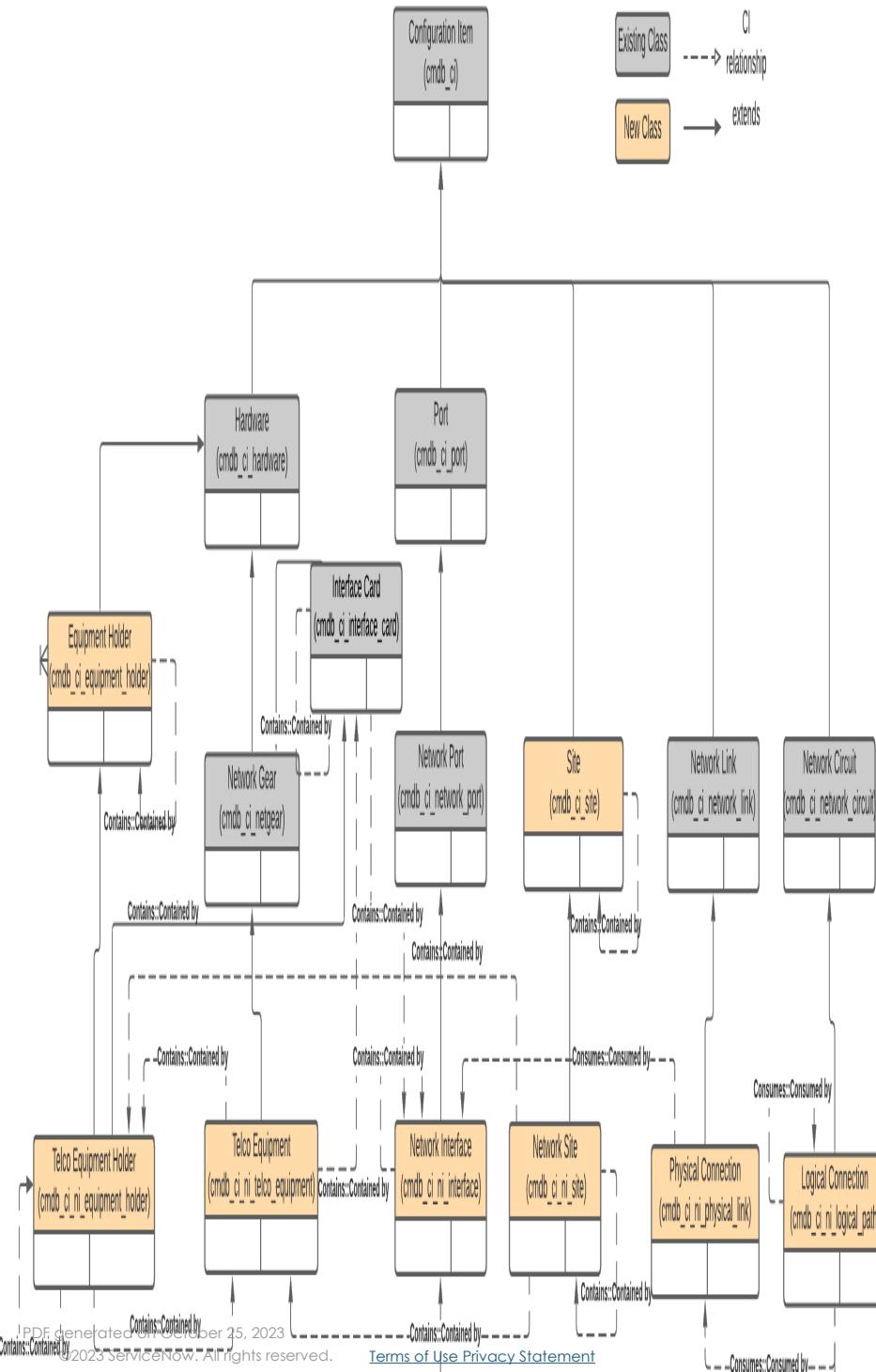
- Class descriptions
- Identification rules
- Identifier entries
- Dependent relationships, if applicable.

With the ServiceNow Telecommunications Network Inventory application, you can build a digital representation of your physical and logical networks. This application uses the NI class extensions to populate CIs that form the basis of your digital network inventory model. To learn more, see [Telecommunications Network Inventory](#)

Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

Telecommunications Network Inventory (NI) schema structure



Classes

This section lists the relevant classes that the CMDB CI Class Models store app adds or updates. See the class columns table for further details about the columns added for each class. For the list of CMDB classes in a base system, including ones that this store app might be extending, see [CMDB tables descriptions](#).

NI extension classes

Class	Extends	Description
Network Site [cmdb_ci_ni_site]	Site [cmdb_ci_site]	<p>Network Site.</p> <p>Captures and maintains the location-specific attributes for each network site, including the network centers, buildings, floors, and rooms where equipment is located.</p> <p>The network site records enable you to view all the equipment at a location. You can filter the locations by the assigned type, role, or function categories.</p>
Telco Equipment Holder [cmdb_ci_ni_equipment_holder]	Equipment Holder [cmdb_ci_equipment_holder]	<p>Telco Equipment Holder.</p> <p>Represents the physical units that contain the telecommunications equipment, including the cages, bays,</p>

Class	Extends	Description
		<p>cabinets, slots, relay racks, and line ups. The line ups contain the individual relay racks. Each relay rack contains the equipment shelves. The equipment holders can contain the other equipment holders.</p> <p>The equipment holder records enable you to track and manage your network assets.</p>
Telco Equipment [cmdb_ci_ni_equipment]	Network gear [cmdb_ci_netgear]	<p>Telco Equipment.</p> <p>Represents a shelf or device that provides the technical functionality in a network. Examples include the routers, modems, mobile devices, optical cables, relays, and switches. The equipment can have slots, cards, or ports. The equipment can exist within an equipment holder or by itself because not all equipment is rack mounted.</p> <p>The equipment records enable you</p>

Class	Extends	Description
		to track and manage your network assets.
Network Interface [cmdb_ci_interface]	Network Port [cmdb_ci_ni_network_port]	Network Interface. Captures and maintain equipment-specific attributes for the network interfaces.
Interface Card [cmdb_ci_ni_interface_card]	Hardware [cmdb_ci_hardware]	Network Interface Card. Represents interface cards that are stored in a network. Cards can occupy more than one slot and can contain other cards. They can be the equipment ports that are physical or logical (virtual). Each port is assigned a bandwidth value. The bandwidths are consumed when used in network designs.
Physical Connection [cmdb_ci_ni_physical_link]	Network Link [cmdb_ci_network_link]	Physical Connection. Represents the physical port connections on the interface cards in your networks.

Class	Extends	Description
Logical Connection [cmdb_ci_ni_logical_path]	Network Circuit [cmdb_ci_network_circuit]	Logical Connection. Represents the logical or virtual port connections on the network interface cards. A logical connection typically represents the multiple physical connections on an interface card.

Nutanix extension classes

The CMDB CI Class Models store app adds or updates classes for Nutanix.

The app adds class models that extend the CMDB class hierarchy, including class descriptions, identification rules, identifier entries, and dependent relationships if applicable. You can use the added classes as any other CMDB class. Applications such as Discovery and Service Mapping patterns can use these class extensions to populate CIs and discover various technologies and software.

Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

Nutanix Enterprise Cloud platform

The Nutanix Enterprise Cloud platform is a converged, scale-out compute and storage system that hosts and stores virtual machines. All nodes in a Nutanix cluster share the management of cluster resources. The foundational unit for the cluster is a Nutanix node which runs a standard hypervisor and contains processors, memory, and local storage (SSDs and hard disks). A Nutanix Controller virtual machine runs on each node, enabling the pooling of local storage from all nodes in the cluster.

Classes

This section lists the classes that the CMDB CI Class Models store app adds or updates. See the class columns table for further details about the columns added for each class.

CMDB CI Class Models: Release 1.6.0 adds the following classes for Nutanix. For the list of CMDB classes in a base system, including ones that this store app might be extending, see [CMDB tables descriptions](#).

Class	Extends	Description
Nutanix Prism Central [cmdb_ci_nutanix_prism_central]	Virtual Machine Object [cmdb_ci_vm_object]	Multi-cluster manager responsible for managing multiple Acropolis Clusters to provide a single, centralized management interface.

CMDB CI Class Models: Release 1.2.0 adds the following classes for Nutanix.

Class	Extends	Description
Nutanix Cluster [cmdb_ci_nutanix_cluster]	Virtual Machine Object [cmdb_ci_vm_object]	Cluster comprising of the physical nodes running Nutanix software.
Nutanix Controller VM [cmdb_ci_nutanix_controller_vm]	Application [cmdb_ci_appl]	Nutanix controller virtual machine that is present in each node and that provides the storage clustering and management capabilities.
Nutanix Storage Container	Storage Volume	Subset of Nutanix storage pool used to apply policies such

Class	Extends	Description
[cmdb_ci_nutanix_storage_container]	[cmdb_ci_storage_volume]	as reserved capacity, replication factor, and storage optimization options.
Nutanix Storage Pool [cmdb_ci_nutanix_storage_pool]	Storage Pool [cmdb_ci_storage_pool]	Grouping of physical disks within a Nutanix cluster which is typically used to create physical separation between virtual machines.
Nutanix Host [cmdb_ci_nutanix_host]	Virtualization Server [cmdb_ci_virtualization_server]	Physical host on which all the virtual machines run.
Nutanix Virtual Machine Instance [cmdb_ci_nutanix_vm_instance]	Virtual Machine Instance [cmdb_ci_vm_instance]	A virtual machine that runs on Nutanix infrastructure.

CMDB CI Class Models: Release 1.1.5 adds the following classes for Nutanix.

Class	Extends	Description
Nutanix Cluster [cmdb_ci_nutanix_cluster]	Virtual Machine Object [cmdb_ci_vm_object]	Cluster comprising of the physical nodes running Nutanix software.
Nutanix Controller VM [cmdb_ci_nutanix_controller_vm]	Application [cmdb_ci_appl]	Nutanix controller virtual machine that is present in each node and that provides the storage clustering

Class	Extends	Description
		and management capabilities.
Nutanix Storage Container [cmdb_ci_nutanix_storage_container]	Storage Volume [cmdb_ci_storage_volume]	Subset of Nutanix storage pool used to apply policies such as reserved capacity, replication factor, and storage optimization options.
Nutanix Storage Pool [cmdb_ci_nutanix_storage_pool]	Storage Pool [cmdb_ci_storage_pool]	Grouping of physical disks within a Nutanix cluster which is typically used to create physical separation between virtual machines.
Nutanix Host [cmdb_ci_nutanix_host]	Virtualization Server [cmdb_ci_virtualization_server]	Physical host on which all the virtual machines run.
Nutanix Virtual Machine Instance [cmdb_ci_nutanix_vm_instance]	Virtual Machine Instance [cmdb_ci_vm_instance]	A virtual machine that runs on Nutanix infrastructure.

CMDB CI Class Models: Release 1.1.4 adds the following classes for Nutanix.

CMDB CI Class Models: Release 1.1.4 adds the following classes for Nutanix.		
Nutanix Cluster [cmdb_ci_nutanix_cluster]	Virtual Machine Object [cmdb_ci_vm_object]	Cluster comprising of the physical nodes running Nutanix software.

Nutanix Controller VM [cmdb_ci_nutanix_controller_vm]	Application [cmdb_ci_appl]	Nutanix controller virtual machine that is present in each node and that provides the storage clustering and management capabilities.
Nutanix Storage Container [cmdb_ci_nutanix_storage_container]	Storage Volume [cmdb_ci_storage_volume]	Subset of Nutanix storage pool used to apply policies such as reserved capacity, replication factor, and storage optimization options.
Nutanix Storage Pool [cmdb_ci_nutanix_storage_pool]	Storage Pool [cmdb_ci_storage_pool]	Grouping of physical disks within a Nutanix cluster which is typically used to create physical separation between virtual machines.
Nutanix Host [cmdb_ci_nutanix_host]	Virtualization Server [cmdb_ci_virtualization_server]	Physical host on which all the virtual machines run.
Nutanix Virtual Machine Instance [cmdb_ci_nutanix_vm_instance]	Virtual Machine Instance [cmdb_ci_vm_instance]	A virtual machine that runs on Nutanix infrastructure.

Class columns

CMDB CI Class Models: Release 1.6.0 adds no columns to the Nutanix Prism Central [cmdb_ci_nutanix_prism_central] class.

CMDB CI Class Models: Release 1.2.0 adds the following columns to the respective classes.

Nutanix Cluster [cmdb_ci_nutanix_cluster] class

Added columns	Description
Block Serial Numbers	Serial numbers of blocks that are connected to the cluster.
Cluster ID	UUID (Universal Unique Identifier) of the cluster.
External Subnet	Subnet of the external IP address of the cluster.
Full Version	Full version of the cluster. For example: el7.3-release-euphrates-5.10.3.1-stable-655d4def34bf18785782f2ad b8cdd5f8457d1fe3
Hypervisor Types	Types of hypervisors that are related to this cluster.
Internal Subnet	Subnet of internal IP addresses.
NCC Version	Nutanix cluster check version.
NTP Servers	NTP servers that are related to this cluster.
Number of Nodes	Number of nodes that are connected to the cluster.
Timezone	Timezone of the cluster.
Version	Version of the cluster.

Added columns	Description
	For example: 5.10.3.1

Nutanix Controller VM [cmdb_ci_nutanix_controller_vm] class

Added columns	Description
Hypervisor Type	Type of hypervisor.
Memory (MB)	Amount of memory (in MB) available on the controller.
State	On/off power state of controller.
VM ID	UUID of the controller virtual machine.
Object ID	ID of the controller virtual machine.

Nutanix Storage Container [cmdb_ci_nutanix_storage_container] class

Added columns	Description
Compression	Indicates whether compression is enabled.
Container ID	UUID of the container.
Deduplication	<p>Indicates whether on disk deduplication is enabled, that is dedup compression applied to data on hard disks (HDD).</p> <p>Performance tier deduplication is a prerequisite for on disk deduplication.</p>

Added columns	Description
Erasure Code	Indicates whether erasure coding is enabled.
Replication Factor	Number of maintained data copies. The replication factor is specified (2 or 3) when the container is created.

The following classes have no added columns:

- Nutanix Storage Pool [cmdb_ci_nutanix_storage_pool]
- Nutanix Host [cmdb_ci_nutanix_host]
- Nutanix Virtual Machine Instance [cmdb_ci_nutanix_vm_instance]

CMDB CI Class Models: Release 1.1.5 adds the following columns to the respective classes.

Nutanix Cluster [cmdb_ci_nutanix_cluster] class

Added columns	Description
Block Serial Numbers	Serial numbers of blocks that are connected to the cluster.
Cluster ID	UUID (Universal Unique Identifier) of the cluster.
External Subnet	Subnet of the external IP address of the cluster.
Full Version	Full version of the cluster. For example: el7.3-release-euphrates-5.10.3.1-stable-655d4def34bf18785782f2ad b8cdd5f8457d1fe3

Added columns	Description
Hypervisor Types	Types of hypervisors that are related to this cluster.
Internal Subnet	Subnet of internal IP addresses.
NCC Version	Nutanix cluster check version.
NTP Servers	NTP servers that are related to this cluster.
Number of Nodes	Number of nodes that are connected to the cluster.
Timezone	Timezone of the cluster.
Version	<p>Version of the cluster. For example: 5.10.3.1</p>

Nutanix Controller VM [cmdb_ci_nutanix_controller_vm] class

Added columns	Description
Hypervisor Type	Type of hypervisor.
Memory	Amount of memory (in MB) available to the virtual machine.
State	On/off state of power.
VM ID	UUID of the controller virtual machine.
Object ID	ID of the controller virtual machine.

Nutanix Storage Container [cmdb_ci_nutanix_storage_container] class

Added columns	Description
Compression	Indicates whether compression is enabled.
Container ID	UUID of the container.
Deduplication	Indicates whether on disk deduplication is enabled, that is dedup compression applied to data on hard disks (HDD). Performance tier deduplication is a prerequisite for on disk deduplication.
Erasure Code	Indicates whether erasure coding is enabled.
Replication Factor	Number of maintained data copies. The replication factor is specified (2 or 3) when the container is created.

The following classes have no added columns:

- Nutanix Storage Pool [cmdb_ci_nutanix_storage_pool]
- Nutanix Host [cmdb_ci_nutanix_host]
- Nutanix Virtual Machine Instance [cmdb_ci_nutanix_vm_instance]

CMDB CI Class Models: Release 1.1.4 adds the following columns to the respective classes.

Nutanix Cluster [cmdb_ci_nutanix_cluster] class

Added columns	Description
Block Serial Numbers	Serial numbers of blocks that are connected to the cluster.
Cluster ID	UUID (Universal Unique Identifier) of the cluster.
External Subnet	Subnet of the external IP address of the cluster.
Full Version	Full version of the cluster. For example: el7.3-release-euphrates-5.10.3.1-stable-655d4def34bf18785782f2ad b8cdd5f8457d1fe3
Hypervisor Types	Types of hypervisors that are related to this cluster.
Internal Subnet	Subnet of internal IP addresses.
NCC Version	Nutanix cluster check version.
NTP Servers	NTP servers that are related to this cluster.
Number of Nodes	Number of nodes that are connected to the cluster.
Timezone	Timezone of the cluster.
Version	Version of the cluster. For example: 5.10.3.1

Nutanix Controller VM [cmdb_ci_nutanix_controller_vm] class

Added columns	Description
Hypervisor Type	Type of hypervisor.
Memory	Amount of memory (in MB) available to the virtual machine.
State	On/off state of power.
VM ID	UUID of the controller virtual machine.
Object ID	ID of the controller virtual machine.

Nutanix Storage Container [cmdb_ci_nutanix_storage_container] class

Added columns	Description
Compression	Indicates whether compression is enabled.
Container ID	UUID of the container.
Deduplication	Indicates whether on disk deduplication is enabled, that is dedup compression applied to data on hard disks (HDD). Performance tier deduplication is a prerequisite for on disk deduplication.
Erasure Code	Indicates whether erasure coding is enabled.
Replication Factor	Number of maintained data copies. The replication factor is

Added columns	Description
	specified (2 or 3) when the container is created.

The following classes have no added columns:

- Nutanix Storage Pool [cmdb_ci_nutanix_storage_pool]
- Nutanix Host [cmdb_ci_nutanix_host]
- Nutanix Virtual Machine Instance [cmdb_ci_nutanix_vm_instance]

Related concepts

- [CMDB schema model](#)

OpenStack extension classes

The CMDB CI Class Models store app adds or updates classes for OpenStack.

The app adds class models that extend the CMDB class hierarchy, including class descriptions, identification rules, identifier entries, and dependent relationships if applicable. You can use the added classes as any other CMDB class. Applications such as Discovery and Service Mapping patterns can use these class extensions to populate CIs and discover various technologies and software.

Request apps on the Store

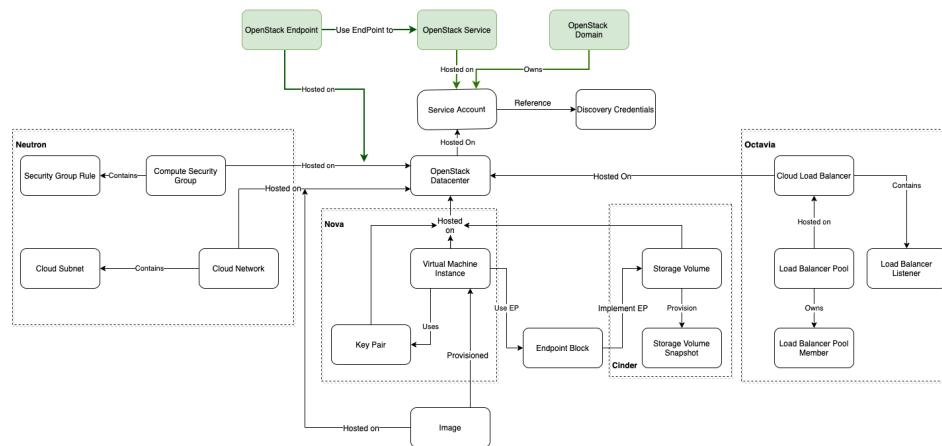
Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

OpenStack

OpenStack is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter. All of these resources are managed and provisioned through APIs with common authentication mechanisms. Other components provide services such as orchestration, fault management, and service management to ensure high availability of user applications. OpenStack

is broken up into services to enable you to plug and play components depending on your needs. These components are designed for horizontal scalability, so you can easily add new resources to grow your cloud over time.

OpenStack classes integrated with the CMDB class hierarchy



Classes

This section lists the relevant classes that the CMDB CI Class Models store app adds or updates. See the class columns table for further details about the columns added for each class.

CMDB CI Class Models: Release 1.8.0 adds the following classes for OpenStack. For the list of CMDB classes in a base system, including ones that this store app might be extending, see [CMDB tables descriptions](#).

Class	Extends	Description
OpenStack Services [cmdb_ci_cloud_open_stack_service]	Virtual Machine Object [cmdb_ci_vm_object]	An OpenStack web service that can be accessed via a URL.
OpenStack Endpoint [cmdb_ci_cloud_open_stack_endpoint]	Virtual Machine Object [cmdb_ci_vm_object]	The access point of a Service.

Class	Extends	Description
OpenStack Domain [cmdb_ci_cloud_openstack_domain]	Virtual Machine Object [cmdb_ci_vm_object]	A collection of users, groups, and projects.

Class columns

CMDB CI Class Models: Release 1.8.0 adds the following columns to the respective classes.

OpenStack Services [cmdb_ci_cloud_openstack_service] class

Added columns	Description
type	The Service type, which describes the API implemented by the Service. Possible values: Compute, ec2, identity, image, network, or volume.
enabled	Defines if the service and its endpoints appear in the Service catalog (true/false).

OpenStack Endpoint [cmdb_ci_cloud_openstack_endpoint] class

Added columns	Description
interface	<p>The interface type, which describes the visibility of the endpoint. Possible values:</p> <ul style="list-style-type: none"> • public - Visible by end users on a publicly available network interface. • internal - Visible by end users on an unmetered internal network interface.

Added columns	Description
	<ul style="list-style-type: none">• admin - Visible by administrative users on a secure network interface.
enabled	Defines if the Service and its endpoints appear in the Service catalog (true/false).

OpenStack Domain [cmdb_ci_cloud_openstack_domain] class

Added columns	Description
enabled	Defines if the domain is enabled (true/false).

Related concepts

- [CMDB schema model](#)

Operational Technology (OT) extension classes

The CMDB CI Class Models store app adds or updates classes for Operational Technology (OT).

Operational Technology (OT) data model

The Operational Technology (OT) data model was created to enable management of "OT asset" data. Operational Technology that controls industrial equipment can be based on IT class hardware (computers, servers, network gear, and so on), or on specific hardware profiles not included in the ServiceNow IT class model (PLCs, HMI, Engineering Workstations, Historians, and so on). Therefore, a single OT asset in the OT data model includes two primary components:

1. A CI class record. This can be an IT or an OT class CI.
2. An OT Asset details record. This describes the OT asset type (function) and other OT-specific attributes.

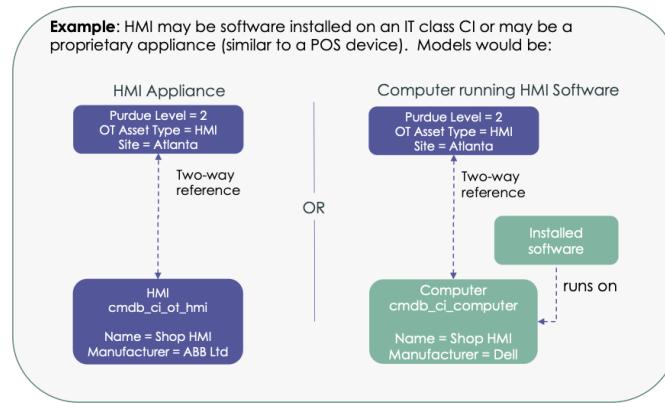
Each OT Asset in the CMDB can be distinguished as having an “OT Asset Details” reference [cmdb_ot_entity reference field] to a specific OT Asset details [cmdb_ot_entity table] record. This is a bi-directional reference; the ot_asset reference on the cmdb_ot_entity table references the CI record. If the cmdb_ot_entity reference of a given CI record is **not** empty, the CI is considered to be an OT asset.

For example, an HMI (Human Machine Interface) OT asset could be composed in at least two different ways.

1. A computer CI with an OT asset detail record describing its “OT Asset type” as HMI.
2. An HMI CI with an OT asset detail record describing its “OT Asset type” also as HMI.

A record in any CI class can be designated as an “OT Asset”

All OT Asset records include a record in the OT Entity table **cmdb_ot_entity** to stores OT specific metadata



In this way, a list of HMIs can be derived and a list of all computers in an enterprise can also be derived.

For more details on the OT data model, see [Operational Technology product view](#).

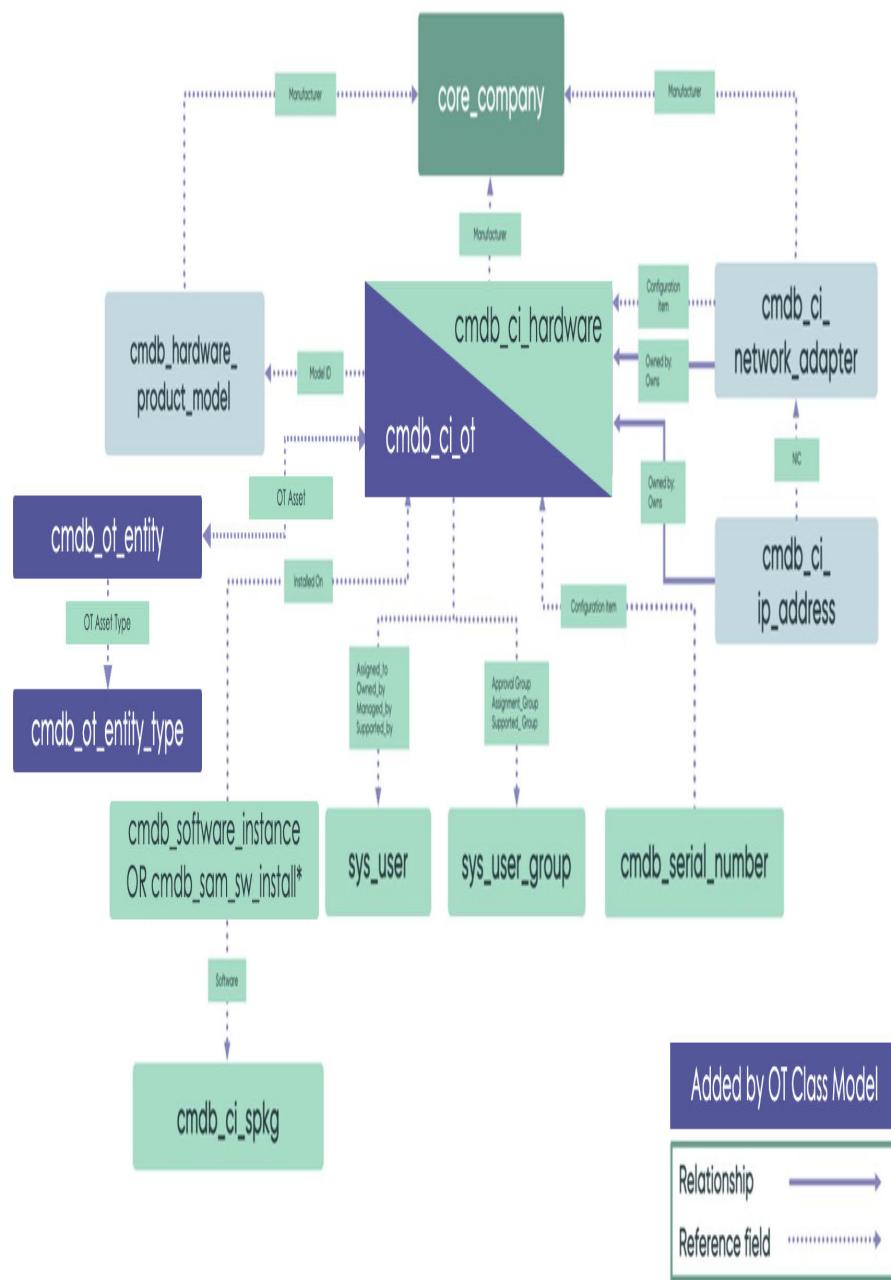
You can use the added classes as any other CMDB class. Applications such as Discovery for Operational Technology, and Service Graph Connector for Operational Technology (Excel) use these class extensions

to populate Cls and discover various technologies and software. To learn more, see:

- [Service Graph Connector for Operational Technology \(Excel\)](#)
- [Discovery for Operational Technology](#)

Note: In Operational Technology, Cls used on an OT network to automate an industrial process are often referred to as OT Assets. This term shouldn't be confused with an asset record commonly used in the practice of Asset Management.

Operational Technology (OT) schema structure



*if software asset management is installed

Classes

This section lists the relevant classes that the CMDB CI Class Models store app adds or updates. See the class columns table for further details about the columns added for each class.

CMDB CI Class Models: Release 1.30 adds the following classes for Operational Technology (OT). For the list of CMDB classes in a base system, including ones that this store app might be extending, see [CMDB tables descriptions](#).

Class	Extends	Description
CNC [cmdb_ci_ot_cnc]	cmdb_ci_ot_control	Computer Numerical Control, used for automated control of machining tools such as drills, lathes, mills, and for 3D printers.
DCS [cmdb_ci_ot_dcs]	cmdb_ci_ot_control	Distributed Control System. Achieves control using intelligence distributed about the controlled process, rather than by a centrally located single unit.
DPU [cmdb_ci_ot_dpu]	cmdb_ci_ot_control	Distributed Processing Units. ICS on a dedicated network, with each DPU handling thousands of points of I/O.
EWS [cmdb_ci_ot_ews]	cmdb_ci_ot_supervisory	Engineering Workstation. A computing platform for configuration, maintenance, and diagnostics of ICS applications and

Class	Extends	Description
		other control system equipment.
Historian [cmdb_ci_ot_historian]	cmdb_ci_ot_supervisory	Data Historian. A centralized database supporting data analysis for industrial processes.
HMI [cmdb_ci_ot_hmi]	cmdb_ci_ot_supervisory	Human-Machine Interface. Hardware or software through which an operator interacts with a controller.
IED [cmdb_ci_ot_ied]	cmdb_ci_ot_control	Intelligent Electronic Device. Receives or sends data/control from or to an external source for power grids.
Industrial Actuator [cmdb_ci_ot_industrial_actuator]	cmdb_ci_ot_field_device	Component of a machine that is responsible for moving and controlling a mechanism, such as opening a valve.
Industrial Drive [cmdb_ci_ot_industrial_drive]	cmdb_ci_ot_field_device	Equipment used to control the speed of machinery. It can be a mechanical, electromechanical, hydraulic, or electronic device.
Industrial 3D Printer [cmdb_ci_ot_industrial_3d_printer]	cmdb_ci_ot_control	Device used in additive manufacturing for the construction of

Class	Extends	Description
		a three-dimensional object from a CAD model, or a digital 3D model.
Industrial Robot [cmdb_ci_ot_industrial_robot]	cmdb_ci_ot_field_device	Robotic system used for manufacturing.
Industrial Sensor [cmdb_ci_ot_industrial_sensor]	cmdb_ci_ot_field_device	Sensor device used to monitor the health of equipment
Network Gear [cmdb_ci_netgear]	cmdb_ci_hardware	Network gear is an electronic device which is required for communication and interaction between devices on a computer network.
Operations Technology [cmdb_ci_ot]	cmdb_ci_hardware	Base class for Operational Technology, used for industrial control. For instance, in manufacturing.
OPC Client [cmdb_ci_ot_opc_client]	cmdb_ci_ot_supervisory	Software module that enables applications to acquire data from an OPC Server or conduct supervisory control using an OPC Server.
OPC Server [cmdb_ci_ot_opc_server]	cmdb_ci_ot_control	Software module that enables applications to provide their data to the outside world using OPC.

Class	Extends	Description
OT Control Module [cmdb_ci_ot_control_module]	cmdb_ci_ot_control	Module such as a PLC or DCS connected to an OT Control System.
OT Control System [cmdb_ci_ot_control]	cmdb_ci_ot	Base Class for industrial control systems (ICS), usually at Purdue Model Level 1 or 2.
OT Supervisory System [cmdb_ci_ot_supervisory]	cmdb_ci_ot	Base class for supervisory systems, usually at Purdue Model Level 2 or 3
PLC [cmdb_ci_ot_plc]	cmdb_ci_ot_control	Programmable Logic Controller. Used to control OT devices.
Protocol Converter [cmdb_ci_protocol_converter]	cmdb_ci_hardware_network_gear	Device used to convert standard or proprietary protocol of one device to the protocol suitable for the other device or tools to achieve the interoperability.
RTU [cmdb_ci_ot_rtu]	cmdb_ci_ot_control	Remote Terminal Unit. Special purpose data acquisition and control unit designed to support DCS and SCADA remote stations
SCADA Client [cmdb_ci_ot_scada_client]	cmdb_ci_ot_supervisory	Supervisory Control and Data Acquisition. Client that enables an operator to manage a SCADA server.

Class	Extends	Description
SCADA Server [cmdb_ci_ot_scada_se rver]	cmdb_ci_ot_control	Supervisory Control and Data Acquisition. System capable of gathering and processing data and applying operational controls over long distance.

Class columns

CMDB CI Class Models: Release 1.30 adds the following columns to the respective classes.

Operational Technology (OT) [cmdb_ci_ot] class

Added columns	Description
firmware_version	Firmware version reported by discovery source.
hardware_version	Hardware version reported by discovery source.

OT Control System [cmdb_ci_ot_control_system] class

Added columns	Description
has_module	true/false value describing if the system has modules, such as chassis / blade architecture.
backplane_name	System-reported string name or number for the backplane.
backplane_id	System-reported unique ID for the backplane

OT Control Module [cmdb_ci_ot_control_module] class

Added columns	Description
slot_number	Reported slot in the control system this module is using.
module_type	Module type reported by the discovery source.

OT Field Device [cmdb_ci_ot_field_device] class

Added columns	Description
device_type	List that describes if the device provides input, output, or both to the parent control system.

Form view

All Operational Technology (OT) extension classes have a "Default view" form view that includes the OT Asset Details attribute at the top of the form. The following table lists the other classes that have the **Operational Technology (OT)** view on their form context menu. This is the default form view for users with the cmdb_ot_viewer role for the following classes.

Class	Description
base hardware [cmdb_ci_hardware]	Base class for hardware.
base computer [cmdb_ci_computer]	An extension of the Hardware table, capturing computer properties.
base server [cmdb_ci_server]	Base class for all types of servers.
linux server [cmdb_ci_linux_server]	Server running Linux software.
windows server [cmdb_ci_win_server]	A server running Microsoft Windows Server operating system.

Class	Description
IoT Device base class [cmdb_ci_iot]	Parent table that contains Internet of Things device types.
IP Firewall and extended classes [cmdb_ci_ip_firewall]	Contains all network firewalls.
IP Router [cmdb_ci_ip_router]	Specialization of the Network Gear [cmdb_ci_netgear] table.
IP Switch [cmdb_ci_ip_switch]	Specialization of the Network Gear [cmdb_ci_netgear] table.
Protocol Converter [cmdb_ci_protocol_converter]	Device used to convert standard or proprietary protocol of one device to the protocol suitable for the other device or tools to achieve the interoperability.

Classes not included in the table do not have the Operational Technology (OT) view by default. For any additional classes required, you can add the Operational Technology (OT) view to the form context menu. For more information about form context menu options, see [Form context menu](#).

Roles and Access Control Logic (ACLs)

The following roles are associated with the Operational Technology (OT) [cmdb_ci_ot] classes and associated tables that follow:

Admin (cmdb_ot_admin)

Can create, read, update, and delete OT records. An admin can also edit the OT Asset Type on the OT Asset form, and manage specific configurations on the OT entity tables.

Editor (cmdb_ot_editor)

Can create, read, update, and delete OT asset records.

Viewer (cmdb_ot_viewer)

Can read OT asset records.

Note: For the cmdb_ci_ot and cmdb_ot_entity tables:

- A user must have one of these three OT roles to view OT assets.
- IT users with an assigned itil role are restricted from viewing OT assets in the cmdb_ci_ot table, and records in the cmdb_ot_entity table. IT users are still able to see IT asset classes, such as Computer, Installed Software.

OT customers may want to restrict access to OT assets from users with an IT (itil) only role, for both a viewer and an admin role.

Key relationship structures

Use the following key relationships as important guidelines when creating Operational Technology (OT) CIs:

OT Entity

Since any CI class may be found on an OT network, the OT Entity table [cmdb_ot_entity] captures additional attributes required in an OT Environment:

Attribute	Description
Business criticality	Business criticality assigned in the discovery source.
OT Asset	Reference to the CI that is on the OT network.
OT Asset type	<p>The function of the OT asset, regardless of CI class..</p> <ul style="list-style-type: none">• For a dedicated HMI appliance, the CI is in the cmdb_ci_ot_hmi class, and the OT Asset Type should be HMI.• For a computer performing the function of an HMI, the CI is in the cmdb_ci_computer class

Attribute	Description
	and the OT Asset type should be HMI
Purdue level	Purdue level of the OT asset.
Zone	Zone assigned to the OT asset, usually used in the context of a zone/conduit model.

The OT Entity is a related list added to the Operational Technology (OT) [cmdb_ci_ot] table and extended tables. If you want to view OT entity metadata on an existing CI class, first add the related list to the form.

OT Entity Type

The [cmdb_ot_entity_type] table tracks the type of OT asset that an OT or non-OT CI is performing the function of. It serves as a necessary part of the Purdue level data model, and extends the Application File [sys_metadata] table.

The Now Platform includes records representing common OT Asset types of OT Asset CIs. The OT Entity Type table attributes include:

Attribute	Description
Label	Display name of a specific OT asset type.
Name	Value used to identify a specific OT asset type.
OT table	Value used for auto-populating the Type field for the OT entity record [cmdb_ot_entity] of an OT Class CI.
Parent	Parent type of a specific type, which is a reference to

Attribute	Description
	a record in the same table [cmdb_ot_entity_type].

Serial number

During CI identification, the Identification and Reconciliation Engine (IRE) processes search for a serial number in two locations. One is the CI serial number attribute, and the other is the Serial Number [cmdb_serial_number] table, with reference back to the Operational Technology (OT) [cmdb_ci_ot] table.

- It stores any serial number of any type other than system serial number, and only in the Serial Number table (and not in the server CI attribute).
- If the system serial number is available, it stores it in both the Serial Number attribute of the CI and in the Serial Number table.
- The Serial Number table is a many-to-one relationship, linking back to the Operational Technology (OT) CI. This table has a Type field for specifying the type of the serial number.
- For Operational Technology (OT), use the string value system for the serial number type to ensure proper reconciliation across various sources.

Network adapter

Use the Network Adapter [cmdb_ci_network_adapter] class to store network adapters.

1. Set the MAC Address attribute to be the MAC address value.
 - Format the string with colon separators between octets and lower case hexadecimal characters with padded zeros.
 - For example, 'f8:f2:1e:00:d4:66'.
2. Set the Name attribute in the Network Adapter class to be the same as the MAC Address.
3. In the CI Relationship [cmdb_rel_ci] table, create an Owned By:Owns relationship to the associated Hardware CI.
4. Using the CI with a reference to the associated Hardware CI, specify a reference from the Network Adapter [cmdb_ci_network_adapter] table.

IP Address

Use the IP Address [cmdb_ci_ip_address] class to store IP addresses.

1. Store an IP address value in the IP Address attribute, and in the Name attribute to avoid empty Name attributes.
 - Store an IPv4 IP address value using the format 'NNN.NNN.NNN.NNN', with decimal-based octets and period separators. Non-conforming values should be considered invalid and cleansed to null values.
 - Store an IPv6 IP address value using lower case hexadecimal with colon separators. Non-confirming values should be considered invalid and cleansed to null values.
2. Set the Netmask attribute to the IP address.
3. Set the Name attribute in the Network Adapter class to be the same as the MAC Address.
4. In the CI Relationship [cmdb_rel_ci] table, create an Owned By::Owns relationship to the associated Hardware CI.
5. In the CI Relationship [cmdb_rel_ci] table, create an Owned By::Owns relationship to the associated Hardware CI.
6. For the IP address, specify a reference to the Network Adapter [cmdb_ci_network_adapter] table, using the Configuration Item with a reference to the associated Hardware CI.
7. To ensure that base system identification rules work properly, also store the IP address in the associated Network Adapter class.

Network adapter and IP address

Use the IP Address [cmdb_ci_ip_address] class to store IP addresses.

1. Store the MAC address of the network adapter installed on a server, in the Network Adapter [cmdb_ci_network_adapter] class.
2. Store the IP address in the IP Address [cmdb_ci_ip_address] class.

Note: Do not store the MAC address or the IP address in the Operational Technology (OT) [cmdb_ci_ot] table. The default Operational Technology (OT) form is configured to display the IP address from the Network Adapter table.

Key reference structures

Use the following key references are important guidelines when creating Operational Technology (OT) records:

- When creating computer or server records for OT assets that are running on computers or servers, see the following topics:
 - Computer [cmdb_ci_computer] class
 - Server [cmdb_ci_server] class
- The Manufacturer and Model ID attributes are reference attributes to the Company [core_company] and Product Model [cmdb_model] tables respectively.
- The Owned By, Assigned To, Managed By, and Supported By attributes are reference attributes to the User [sys_user] table. The Change Group and Support Group attributes are reference attributes to the Group [sys_user_group] table.

Identification rules

The Now Platform contains a pre-defined identification rule for the Operational Technology (OT) classes. That identification rule has the following key identifier entries, listed in priority order:

1. Identifier entry that uses the identification specified in Serial Number [cmdb_serial_number] as the lookup table. The Serial Number table is a many-to-one reference from the serial number back to the server CI.
2. Identifier entry that is specified in the Serial Number attribute in the CI.
3. Identifier entry that is specified in the Mac Address attribute in the Network Adapter table.
4. Identifier entry for the Name attribute.
 - If Serial Number and MAC Address are not available, then the Name (which is usually the system reported hostname) attribute is used.
 - If both Serial Number and Name are not available, and only MAC Address is available, use MAC Address as the name of the

CI. Using the MAC Address as the name of the CI ensures that you don't create an empty CI.

Note: To learn more, see [CMDB Identification and Reconciliation](#).

Deprecated classes

CMDB CI Class Models: Release 1.30 soft deprecated (not removed, but marked as follows) with the release of the OT class model:

Class	Description
Human Machine Interface [cmdb_ci_hmi]	OT assets moved to cmdb_ci_ot to support broader use cases. Use cmdb_ci_ot_hmi instead.
Manufacturing Device [cmdb_ci_manufacturing]	OT assets moved to cmdb_ci_ot to support broader use cases. Use cmdb_ci_ot as the base class or other generic child classes as appropriate.
Programmable Logic Controller [cmdb_ci_plc]	OT assets moved to cmdb_ci_ot to support broader use cases. Use cmdb_ci_ot_plc instead.

A script has been provided to migrate records from these classes to the new class tables. The admin role is required to perform the following tasks:

1. Navigate to **System Definition > Script Includes**
2. Find the record named **OTAssetsMigrationUtils**.
3. Navigate to **Definition > Script Includes > .**
4. In the background window, copy and paste the provided script.
5. Select the appropriate scope and run the script.

Troubleshooting

The following are some troubleshooting tips:

Problem	Suggested resolution
Unable to see OT Asset menu items	Ensure that the logged in user has been assigned the appropriate roles. To learn more, see the preceding Roles section.
Error creating or updating an OT Asset record	Allow only one OT Asset record (cmdb_ot_entity) per CI.

Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

The [CMDB CI Class Models store app](#) adds class models that extend the CMDB class hierarchy, including:

- Class descriptions
- Identification rules
- Identifier entries
- Dependent relationships, if applicable.

Related concepts

- [CMDB schema model](#)

Red Hat Virtualization (RHV) extension classes

The CMDB CI Class Models store app adds or updates classes for Red Hat Virtualization (RHV).

The app adds class models that extend the CMDB class hierarchy, including class descriptions, identification rules, identifier entries, and dependent relationships if applicable. You can use the added classes as any other CMDB class. Applications such as Discovery and Service

Mapping patterns can use these class extensions to populate CIs and discover various technologies and software.

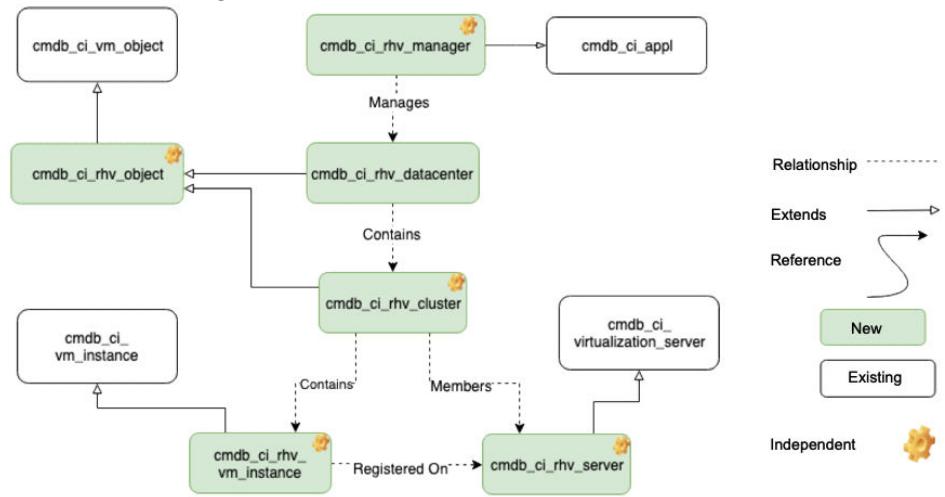
Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

Red Hat Virtualization (RHV)

Red Hat Virtualization (RHV) is a virtualization product which is based on the Kernel-based Virtual Machine (KVM) hypervisor. RHV uses the SPICE protocol and Virtual Desktop Server Manager (VDSM) with a RHEL-based centralized management server. RHV solution is based on two primary software components: Red Hat Virtualization Manager (RHV-M) and Red Hat Virtualization Hypervisors or hosts: Red Hat Enterprise Linux or RHV Host (RHV-H).

RHV classes integrated with the CMDB class hierarchy



Classes

This section lists the relevant classes that the CMDB CI Class Models store app adds or updates. See the class columns table for further details about the columns added for each class.

CMDB CI Class Models: Release 1.8.0 adds the following classes for RHV. For the list of CMDB classes in a base system, including ones that this store app might be extending, see [CMDB tables descriptions](#).

Class	Extends	Description
RHV LDC [cmdb_ci_rhv_ldc]	Logical Datacenter [cmdb_ci_logical_datacenter]	RHV logical datacenter.
RHV Datacenter [cmdb_ci_rhv_datacenter] This class is deleted.	N/A	N/A

CMDB CI Class Models: Release 1.6.0 adds the following classes for RHV.

Class	Extends	Description
RHV Server [cmdb_ci_rhv_server]	Virtualization Server [cmdb_ci_virtualization_server]	The RHV virtualization host.
RHV Object [cmdb_ci_rhv_object]	Virtual Machine Object [cmdb_ci_vm_object]	A base class for other classes to derive from.
RHV Manager [cmdb_ci_rhv_manager]	Application [cmdb_ci_appl]	RHV Manager instance.
RHV Cluster	RHV Object	RHV cluster.

Class	Extends	Description
[cmdb_ci_rhv_cluster]	[cmdb_ci_rhv_object]	
RHV Datacenter [cmdb_ci_rhv_datacenter] Note: This class is being deleted in the CMDB CI Class Models 1.6.0 release.	RHV Object [cmdb_ci_rhv_object]	RHV datacenter.
RHV Virtual Machine Instance [cmdb_ci_rhv_vm_instance]	Virtual Machine Instance [cmdb_ci_vm_instance]	RHV virtual machine instance.

Class columns

CMDB CI Class Models: Release 1.6.0 adds the following columns to the respective classes.

RHV Server [cmdb_ci_rhv_server] class

Added columns	Description
url	URL used to access the object.

RHV Object [cmdb_ci_rhv_object] class

Added columns	Description
href_id	Href ID.
manager_id	Manager ID.

Added columns	Description
url	URL to access the object (used in child classes).

RHV Manager [cmdb_ci_rhv_manager] class

Added columns	Description
url	URL of the manager.
product_name	Product name.

RHV Cluster [cmdb_ci_rhv_cluster] class

Added columns	Description
cpu_type	CPU type.
cpu_architecture	CPU architecture.
ksm	Enabled state of kernel same-page merging (KSM) memory policy.
fencing_policy	Fencing policy.
memory_overcommitment	Amount of over-commitment memory allowed on the cluster.
transparent_huge_pages	Transparent huge memory pages policy.
ballooning	Memory ballooning for guests.
compatibility_version	Compatibility version.

RHV Virtual Machine Instance [cmdb_ci_rhv_vm_instance] class

Added columns	Description
mgmt_url	Management URL.

Added columns	Description
cpu_architecture	CPU architecture.
delete_protected	Delete protected (true/false).
ha_priority	HA priority.
high_availability	High availability (true/false).
memory_policy_guaranteed	Amount of memory guaranteed (MB).
memory_policy_max_mb	Maximum memory in the dynamic memory allocation policy.
multi_queues	Multi queues.
placement_policy	Placement policy.
cpu_sockets	Number of CPU sockets.
stateless	Stateless (true/false).
storage_error_resume_behaviour	Behavior of a virtual machine that is paused due to storage I/O error. For examples, AUTO_RESUME, LEAVE_PAUSED, and KILL.
start_time	Start time.
stop_time	Stop time.
threads	Number of threads.
time_zone	Time zone.
usb	USB enabled state.
run_once	Run once.
type	RHV type.

The following class was deleted in the CMDB CI Class Models 1.6.0 release.

RHV Datacenter [cmdb_ci_rhv_datacenter] class

Added columns	Description
quota_mode	Quota mode policy.

Related concepts

- [CMDB schema model](#)

Transport Layer Security (TLS) extension classes

The CMDB CI Class Models store app adds or updates a class for TLS certificates.

The app adds class models that extend the CMDB class hierarchy. You can use the added classes as any other CMDB class. Applications such as Discovery and Service Mapping patterns can use these class extensions to populate CIs and discover various technologies and software.

Request apps on the Store

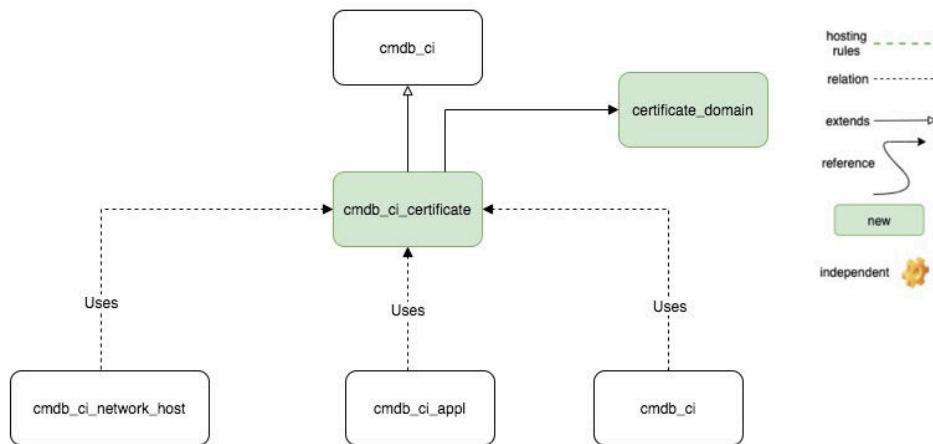
Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

Transport Layer Security (TLS)

TLS is a cryptographic protocol designed to provide communications security over a computer network. The TLS protocol provides privacy and data integrity between communicating computer applications. Once the client and the server have agreed to use TLS, they negotiate a stateful connection by using a handshaking procedure. The server usually provides identification in the form of a digital certificate. The certificate contains the server name, the trusted certificate authority (CA) that vouches for the authenticity of the certificate, and the server's public encryption key. The client confirms the validity of the certificate before

proceeding. When the handshake is completed, a secured connection is established.

TLS certificate classes integrated with the CMDB class hierarchy



Scoped apps certification class

The scoped apps certification class supports TLS certificates. With this class you can proactively manage certificates by keeping stakeholders informed about any impending expiries. Use this extension class to ensure that certificates are monitored and renewed before they expire, to prevent severe outage of production systems.

Classes

This section lists the relevant classes that the CMDB CI Class Models store app adds or updates. See the class columns table for further details about the columns added for each class.

The CMDB CI Class Models: Release 1.4.0 updates the following class:

Class	Extends	Description
Unique Certificate [cmdb_ci_certificate]	Configuration Item [cmdb_ci]	A public key certificate in X.509 standard format.

The CMDB CI Class Models store app changes the Unique Certificate [cmdb_ci_certificate] class as follows:

- The assigned_to attribute now depends on the assignment_group attribute so that users in the assigned_to attribute are filtered based on the specified assignment_group.
- The [Certificate Inventory and Management](#) store app populates the Unique Certificate [cmdb_ci_certificate] table. The list view for that class does not have a **New** button and you can no longer add new records to the table. This is because there are certain fields that are extracted from binary encoded parameters in the certificate which users may not be able to provide. Also, certificates have to be discovered rather uploaded.
- You can no longer add or delete attachments in the Certificate file attribute.

The CMDB CI Class Models: Release 1.3.0 adds the following classes. For the list of CMDB classes in a base system, including ones that this store app might be extending, see [CMDB tables descriptions](#).

Class	Extends	Description
Unique Certificate [cmdb_ci_certificate]	Configuration Item [cmdb_ci]	N/A
Certificate Domain [certificate_domain]	N/A	Fully qualified domain name.

Class columns

CMDB CI Class Models: Release 1.4.0 adds the following columns to the respective classes.

Unique Certificate [cmdb_ci_certificate] class

Added columns	Description
Certificate file	Certificate in an encoded form.
Fingerprint	Hash value of the certificate.
Fingerprint algorithm	Algorithm used to hash the certificate.

Added columns	Description
Is certificate authority	Indicates whether a certificate is a Certificate Authority (CA) or not.
Is selfsigned	Indicates whether the certificate is self-signed or not.
Issuer	Entity that has signed and issued the certificate. Reference: Unique Certificate [cmdb_ci_certificate]
Issuer common name	Common name of the issuer.
Issuer distinguished name	Distinguished name of the issuer.
Key size	Size of the key used by the signing algorithm. Choices: <ul style="list-style-type: none">• Create priority 1 tasks• Create priority 3 tasks• Do not create renewal tasks
Renewal tracking	Indicates whether to create any priority 1 or priority 3 tasks for the expiring certificates.
Root issuer	Root entity that has signed and issued the intermediate certificate. Choices: <ul style="list-style-type: none">• External• Internal

Added columns	Description
	Reference: Unique Certificate [cmdb_ci_certificate]
Service type	Indicates whether the certificate is used for external or internal services.
Signature algorithm	<p>The cryptographic algorithm used to sign the certificate. Choices:</p> <ul style="list-style-type: none"> • Issued • Installed • Revoked • Retired
State	Lifecycle states of the certificate.
Subject alternative name	<p>List of fully qualified domain names secured by the certificate. Reference: Certificate Domain [certificate_domain]</p>
Subject common name	Identifies the hostname/domain associated with the certificate.
Subject country	Subject's two letter country code.
Subject distinguished name	Identifying information of the subject.
Subject email	Subject's email.
Subject locality	Subject's locality.

Added columns	Description
Subject organization	Subject's organization.
Subject organizational unit	Subject's organizational unit.
Subject state	Subject's state.
Valid from	Validity start period of the certificate.
Valid to	Validity end period of the certificate.
Version	X.509 version of the certificate.

Certificate Domain [certificate_domain] class

Added columns	Description
Domain	Fully qualified domain name.

CMDB CI Class Models: Release 1.3.0 adds no columns.

Related concepts

- [CMDB schema model](#)

VMware NSX load balancer extension classes

The CMDB CI Class Models store app adds or updates classes for VMware NSX load balancers.

The app adds class models that extend the CMDB class hierarchy, including class descriptions, identification rules, identifier entries, and dependent relationships if applicable. You can use the added classes as any other CMDB class. Applications such as Discovery and Service Mapping patterns can use these class extensions to populate CIs and discover various technologies and software.

Request apps on the Store

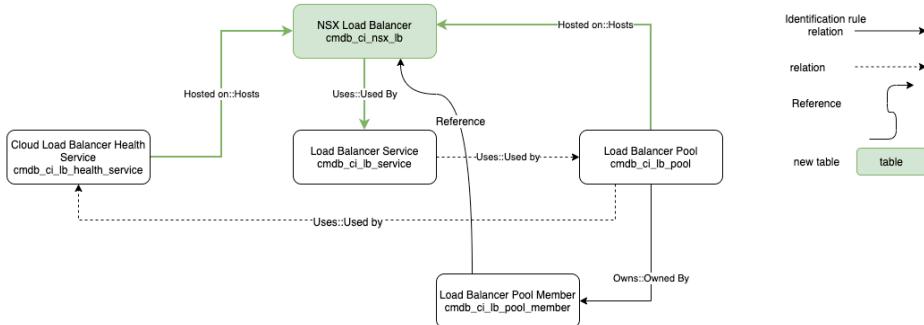
Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

VMware NSX load balancer

NSX is a network virtualization solution offered by VMware. Among the virtual resources included in the NSX solution are virtual LANs (VLANs), virtual load balancers, virtual routers, switches, and firewalls.

ServiceNow Discovery uses the [VMware NSX Advanced load balancer discovery](#) pattern to find VMware NSX load balancers and their components: Listeners, pools, pool members, and health services.

VMware NSX load balancer classes integrated with the CMDB class hierarchy



Classes

This section lists the classes that the CMDB CI Class Models store app adds or updates.

CMDB CI Class Models: Release 1.10.0 adds the following classes for the VMware NSX load balancer. For the list of CMDB classes in a base system, including ones that this store app is extending, see [CMDB tables descriptions](#).

Class	Extends	Description
NSX Load Balancer [cmdb_ci_nsx_lb]	Load Balancer [cmdb_ci_lb]	The table containing the NSX Load Balancer resources.

Class columns

The [VMware NSX Advanced load balancer discovery pattern](#) introduces one new table with one identification rule and entry. The table uses only the columns inherited from its parent.

Related concepts

- [CMDB schema model](#)

Extend classes and rules

Extend and update CMDB CI Class Models store app classes and rules when using third-party integration tools.

When using a third-party tool to integrate with ServiceNow apps, gaps can occur between the integration and different CIs. Some of the integrated tables and classes may be missing classes and rules not included with the CMDB CI Class Models store app.

The classes and rules in the following table enable you to add and extend the CMDB CI Class Models store app for integrations:

Table	Extends	Rules and related entries
Postgresql Schema cmdb_ci_postgresql_schema	cmdb_ci_db_catalog	Containment rule: cmdb_ci_db_postgres ql_instance->Contains:Contained By-

Table	Extends	Rules and related entries
		<p>>cmdb_ci_postgresql_schema</p> <p>Identification Rule:</p> <p>Dependent, Attributes: name</p>
Information Object cmdb_ci_information_object	cmdb_ci	<p>Related Entries (cmdb_related_entry):</p> <ul style="list-style-type: none"> • Identifier: cmdb_ci_postgresql_schema • Related table: cmdb_key_value • Referenced field: configuration_item
		<p>Identification rule:</p> <p>Independent, attributes: name</p> <p>Related Entries (cmdb_related_entry):</p> <ul style="list-style-type: none"> • Identifier: cmdb_ci_information_object • Related table: cmdb_key_value

Table	Extends	Rules and related entries
		<ul style="list-style-type: none"> Referenced field: configuration_item
Oracle Catalog cmdb_ci_db_ora_catalog	cmdb_ci_db_catalog	Related Entries (cmdb_related_entry): <ul style="list-style-type: none"> Identifier: cmdb_ci_db_ora_catalog Related table: cmdb_key_value Referenced field: configuration_item
		Containment rule: cmdb_ci_db_mysql_instance->Contains:Contained By->cmdb_ci_db_mysql_catalog
MySQL Catalog cmdb_ci_db_mysql_catalog	cmdb_ci_db_catalog	Identification rule: Dependent, attributes: name
		Related Entries (cmdb_related_entry): <ul style="list-style-type: none"> Identifier: cmdb_ci_db_mysql_catalog

Table	Extends	Rules and related entries
		<ul style="list-style-type: none"> Related table: cmdb_key_value Referenced field: configuration_item
MS SQL Database cmdb_ci_db_mssql_database	cmdb_ci_db_instance	Related Entries (cmdb_related_entry): <ul style="list-style-type: none"> Identifier: cmdb_ci_db_mssql_database Related table: cmdb_key_value Referenced field: configuration_item
Sybase Catalog cmdb_ci_db_syb_catalog	cmdb_ci_db_catalog	Containment rule: cmdb_ci_db_syb_instance->Contains:Contained By->cmdb_ci_db_syb_catalog
		Identification rule: Dependent attributes: name
		Related Entries (cmdb_related_entry):

Table	Extends	Rules and related entries
		<ul style="list-style-type: none"> Identifier: cmdb_ci_db_syb_catalog Related table: cmdb_key_value Referenced field: configuration_item
DB2 Catalog cmdb_ci_db_db2_catalog	cmdb_ci_db_catalog	Containment rule: cmdb_ci_db_db2_instance->Contains:Contained By->cmdb_ci_db_db2_catalog
		Identification rule: Dependent, attributes: name <p>Related Entries (cmdb_related_entry):</p> <ul style="list-style-type: none"> Identifier: cmdb_ci_db_db2_catalog Related table: cmdb_key_value Referenced field: configuration_item

For the list of CMDB classes in a base system, including ones that this store app is extending, see [CMDB tables descriptions](#).

Related concepts

- [CMDB schema model](#)

Baseline CMDB

CMDB baseline provides capabilities that help you understand and control the changes that have been made to your configuration items (CIs) in the CMDB.

- You can create a baseline, which is a snapshot of your configuration items in the CMDB. You can review the changes that have been made to that configuration item since a previous baseline. Multiple baselines may be created and the system tracks the changes that have been made per baseline.

Creating a baseline captures the attributes of the CI as well as all first-level relationships for the CI. Any changes to the base CI or to any related CIs are captured and displayed. Newly created CIs are not automatically added to a baseline.

- Associate a configuration item with a task, a change or change task, and to propose changes to the CI after the change is complete. You can record changes, and these changes are not applied to the CI immediately but are delayed until the change is complete.

When the change is complete, you can choose to apply the proposed changes which makes all changes previously proposed and associates the changes with the task.

For information about planning and implementing a baseline CMDB, see the [CMDB Baseline life-cycle best practices and Diff Formatter troubleshooting \[KB0829681\]](#) knowledge base article.

Create a CMDB baseline

You can create a baseline for a CI to track updates to the CI over time.

Before you begin

Role required: ecmdb_admin and itil

Procedure

1. Navigate to **All > Configuration > Baselines > Baselines**.
2. Click **New**.
3. Enter a **Name** for the baseline.
By default, the cmdb_ci table is selected so that the record creates the baseline for all configuration items in the system.
4. (Optional) To limit the baseline to specific CIs, select a different **Table** or choose **Conditions** that a CI must meet for it to have a baseline entry.
For example, you might create a baseline for the Database table with the condition **[Location] [is] [<configured location>]**.
5. Click **Submit**.
The creation of a baseline is time consuming and occurs in the background. A message at the top of the record list notifies you that your baseline has been scheduled and you will receive an email when the process is complete.

Display baseline differences

You can see the changes that have been made to a CI or any first level related CIs by configuring the CI form layout to display the CMDB Baseline diff field. This field is labeled Baseline differences on the form.

Before you begin

Role required: itil

About this task

Changes are displayed only for the cmdb_ci table and child tables. You can change the com.cmdb.baseline.max_changes system property to limit the number of relationships and changes that appear in a baseline diff section on a CI form (set to 100 by default).

Procedure

1. Open a CI record.
2. Select the baseline you want to see for this CI from the choice list.
The field displays the details of any changes that were made to the current record for the selected baseline, or indicates that no changes were made.
Details of baseline differences

For: SQL Baseline ▾

Basic attribute changes

2009-09-14 14:46:49 System Administrator - Changed: RAM (MB), Disk space (GB)

RAM (MB): 4 was: -1

Disk space (GB): 500 was: 100

3. To add a relationship to the CI, click the green plus icon in the **Related Items** toolbar.
The new relationship appears below the toolbar. For more information about the Related Items toolbar and how to control the display, see [CI relations formatter](#).
4. Update a related CI and see the changes displayed as **Basic attribute changes** in the current CI record.

Basic Attribute Changes

For: SQL Baseline ▾

Basic attribute changes

2009-09-14 15:17:42 System Administrator - Changed: Sys audit, RAM (MB), Disk space (GB)

: Storage Area Network 002 was: (relationship added) - CI Relationship Change

: ApplicationServerPeopleSoft was: (relationship added) - CI Relationship Change

RAM (MB): 4 was: -1

Disk space (GB): 500 was: 100

What to do next

To improve performance and prevent memory issues when showing large amounts of baseline differences data on CI forms, complete the following steps:

1. Set the system property com.cmdb.baseline.entry.attachment to true.

2. Manually run once the CMDB Baseline convert XML to attachment fix script. For information about running a fix script, see [Run fix scripts](#).

Properties for baseline CMDB

Use the baseline CMDB properties to configure how many changes and relationships for a CI can appear in the baseline diff for the CI.

These properties are available for baseline CMDB. To view and edit these properties, the admin role is required.

Properties for Baseline CMDB

Property	Description
Maximum number of changes and relationships for a CI that can appear in the baseline diff for the CI. <code>com.cmdb.baseline.max_changes</code>	<ul style="list-style-type: none">• Type: integer• Default value: 100• Location: Configuration > CMDB Properties > Baseline Properties
<code>com.cmdb.baseline.entry.attachment</code>	Lets you manually run the CMDB Baseline convert XML to attachment fix script. Setting this property to true and then running the CMDB Baseline convert XML to attachment fix script, enables improved performance when showing large amounts of baseline differences data on CI forms. When false, the CMDB Baseline Diff component uses legacy methods which might fail to load large amounts of baseline differences data. <ul style="list-style-type: none">• Type: true false• Default value: false

Property	Description
	<ul style="list-style-type: none">Location: Add to System Properties [sys_properties] table.Learn more: Baseline CMDB

CMDB Workspace (4.0) store app

The CMDB Workspace is an efficient, central, and modernized way for you to work. Use CMDB Workspace to search and explore the CMDB, examine health and recent activity, and access various CMDB dashboards and tools to support tasks in your organization.

General interaction

- CMDB Workspace leverages many [Performance Analytics](#) capabilities and features, such as indicator sources. Throughout the CMDB Workspace views, you can select the various cards to drill down to Performance Analytics KPI Details panes that show trends for the associated data. On a KPI Details pane, you can modify different settings to change the scope of the data. You can also select **Show Records** to list the records associated with the chart.



- Lists throughout the CMDB Workspace have a filter icon () which you can select to show the filter definition used for the list.

CMDB Workspace doesn't support domain separation.

Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

Note: Starting with the San Diego release, the CMDB Workspace store app is automatically installed when installing or upgrading the product.

Enable demo data

Enable demo data to install demo data-specific scheduled jobs that if the needed requirement is met, generates and populates demo data in CMDB Workspace cards such as:

- Cloud vs Non-cloud resources chart in the CMDB Workspace landing page and the Insights view: If Cloud Service Accounts [cmdb_ci_cloud_service_account] table exists
- Cards in the CMDB 360 view: If CMDB 360 is enabled
- Cards in the Insights view, CMDB Feature Adoption tile:
 - Cls processed by IRE
 - Cls processed by IRE based on source
 - Data Manager
 - Data attestation
 - Query Builder
 - Intelligent Search
- Cards in the What's new tile in the CMDB Workspace landing page.

To populate CMDB Workspace dashboards with demo data:

1. When installing the CMDB Workspace store app, check **Load demo data**.
2. Access the [Demo] — CMDB Workspace demo data scheduled job and select **Execute Now**.

Note: As a general practice, don't enable demo data in a production instance to prevent demo data mixing with real production data.

Prerequisites

- Plugins:
 -

Recommended: CSDM Activation (com.snc.cmdb.csdm.activation)

Allows for legacy Lifecycle Status field mappings and synchronization to legacy status fields. For details about use and customization of life-cycle rules when this plugin isn't activated, see [Life-cycle rules](#).

- Required (activated by default): CMDB CSDM Support (com.snc.cmdb.csdm)
- Required (activated by default): CMDB Page Templates (sn_cmdb_pg_tmplts)
- Required (activated by default): CMDB NLQ Search Connected (sn_cmdb-nlq-search)

For details about activating a plugin, see [Activate a plugin](#).

- Roles: To access the CMDB Workspace, you must, at a minimum, have one of the following roles, which are essential for interacting with the CMDB Workspace. Depending on which of these roles is assigned to you, you might only have access to some of the features available in the CMDB Workspace:

- sn_cmdb_admin
- sn_cmdb_editor
- sn_cmdb_user

Note: As you drill down in the CMDB Workspace, there are some dashboards and list views that require specific roles in addition to the key CMDB Admin, CMDB Editor, or CMDB User roles.

- Features: CMDB Workspace provides access to a wide range of applications and features. However, for CMDB Workspace to provide meaningful reports, overviews, and trends, you must set up and configure some of those features. Setup for such features is listed under Additional requirements.

Access the CMDB Workspace

Navigate to **Workspaces > CMDB Workspace** to access the CMDB Workspace landing page.

In addition to the Home view described below, you can access features in the following views of the CMDB Workspace:

- **Governance view:** Manage tasks such as [attestation tasks](#), that are assigned to you.
- **CMDB 360 view:** View aggregations and analysis of CMDB 360 data on a dashboard, and create CMDB 360 queries.
- **Management view:** View recent key activities and health indicators for the CMDB, and access management tools and dashboards (accessible only to CMDB admins) that support your management tasks.
- **Insights view:** View level of adoption of key CMDB tools, features, and application services. Explore benefits and install those tools and features to maximize the efficiency of your CMDB functionality.

Intelligent Search

Accessible to: CMDB Admin, CMDB Editor, CMDB User.

Lets you use Natural Language Query (NLQ) search capabilities provided by [Intelligent Search for CMDB](#). Use the input field to construct a search string using everyday language. As you type, a dynamic list of relevant suggestions appears, with items matching single words or part phrases in the typed-in text, such as table names.

You can:

- Select **Search tips** to see tips about constructing search strings. See details about the usage, examples for single and multi-table search, advanced filtering, and relationships in Intelligent Search.
- Select **Search** to either run the query if the search string is already fully converted into a valid CMDB query, or to open the Refine your query dialog box.
-

If the search string has no ambiguities with the table name or relationships, then the query runs and the results appear in a list view format.

Only the first 100 results of the query appear in the results pane.

- Select **Load More Results** to view the next set of 100 results.

- Select **Load All Results** to view the rest of the query results, up to the number specified by the `glide.cmdb.query.max_results_limit` system property (10,000 by default).

If the constructed CMDB query contains more than a single table, then the **View in Query Builder** button appears. Select the button to open the **CMDB Query Builder** with your query fully constructed on the Query Builder canvas. You can use the Query Builder to continue editing the query.

- If there are any ambiguities with table names or relationship types in the search string, then the search string can't be converted into a valid CMDB query. In this case, the Refine your query dialog box appears letting you select from suggested CI classes and continue to parse your search string into a valid CMDB query. Those suggested CI classes are based on phrases in your search string. Use the drop-down lists to select the CI classes that match your intended search and then select **View search results** to run the query.
- If Intelligent Search is unable to convert your search string into a valid CMDB query, then selecting **Search** doesn't generate any query results. Instead, a feedback form appears. Fill out the form and select **Submit Feedback** to record your feedback for your CMDB Admin to review.
- Use the Sample searches list to get you started in running a pre-defined search. This list consists of more common searches, or searches that are more difficult to construct such as searches that involve application services.
- Use the Your recent searches list to rerun a previous search.
- Select **Results Feedback** to submit feedback on the search results for your CMDB Admin to review.

For more details about using NLQ with Intelligent Search, see [Intelligent Search for CMDB](#).

Alternatively, you can select **Use conditional search instead** to use a basic `condition builder` functionality where you can specify conditions to search for CIs of a specific class.

You can:

- Select **New condition set** to add a condition phrase.

- Select **Related List Condition** to add a condition phrase for related lists.
- Select **Search** to search through the CMDB.

In the results list, select a CI to see details including a timeline, health overview, and several types of attributes such as key and discovery attributes. For more information, see [CI details pane](#).

UI activity	Additional requirements
<ul style="list-style-type: none">• Mapped Application Service• Application Service• Application Service Group• Dynamic CI Group• Tag Based Service	app_service_user role
<ul style="list-style-type: none">• Business Service• Technical Service• Application Service Outage	service_viewer role

Important actions

Accessible to: CMDB Admin and CMDB Editor.

Important actions of various categories that require your attention or action. There are several task categories such as:

- Health tasks generated by [CMDB Health](#): In health-related cards, such as Duplicate CIs, Orphan CIs, stale CIs, and De-Duplication tasks, select **View CIs** or **View Tasks** to drill down to the list of associated CIs or important tasks.
- Data attestation and life cycle approval tasks generated by the [CMDB Data Manager](#): In data attestation-related cards, such as

Reassignment Requests and Unassigned Overdue cards, select **View Tasks** to drill down to the associated important tasks.

Important actions are stored in the CMDB WS Imp Action Card Config [sn_cmdb_ws_imp_action_card_config] table that is accessible for editing only to users with the sn_cmdb_admin role. Authorized users can modify attributes of an important action such as Active and Filter conditions, but can't modify the Type, Persona, and Table attributes.

- Important action cards show per the logged in user role, as specified in the CMDB WS Imp Action Card Config [sn_cmdb_ws_imp_action_card_config] table.
- A card appears only if there is at least one record that meets the card's filter condition.
- If you drill down a Health-related card and modify an associated CI, any resulting impact to health KPIs might appear only after the next cycle of the [CMDB Health dashboard jobs](#).

For information about managing the cards in the Important actions tile, see [Modify important actions in CMDB Workspace](#).

What's new

Accessible to: CMDB Admin, CMDB Editor, CMDB User.

Counts of newly created CIs within a recent time interval. The New CIs total counts all CI types including applications, hardware, and application services, which also appear in separate cards in the tile. By default, historical data is aggregated for the **Last 24 hours**, which you can set to a different time interval such as **Last 7 days**.

Select a card to drill down to a Performance Analytics KPI Details pane that shows the trend for the respective item.

Counts in the What's new cards are based on the following tables:

Count	Table
New CIs	Configuration Item [cmdb_ci]
New application	Application [cmdb_ci_appl]

Count	Table
New hardware	Hardware [cmdb_ci_hardware]
New application services	Application Service [cmdb_ci_service_auto]

CI overview

The following tabs provide summaries about CIs:

CI Summary

Accessible to: CMDB Admin, CMDB Editor, CMDB User (without tabs).

A chart of all CIs in the CMDB, grouped by up to 20 CMDB groups. The CMDB groups in the chart are specified as groups of closely-related classes.

Select a bar to drill down to the classes in the group and their CIs. Then, drill down a CI to show the [CI details pane](#) with a timeline, health overview, and several types of attributes such as key and discovery attributes.

You can add custom class groups by creating [CMDB groups](#) with the following settings:

- Group type is set to 'CMDB Workspace'.
- Populated by encoded queries.

Such custom class groups will appear after the next time that the CMDB Workspace – Group and Encoded Query Counts scheduled jobs run and update the CMDB Workspace.

Note:

In systems with a very large number of CIs, for example a billion or more CIs, the set of queries that populate the CI Overview widget might run for an unreasonable length of time. In that case, you can choose to switch the default queries with a set of simpler queries that can handle such load and properly load the CI Overview widget. However, the results yielded from the simpler queries aren't as complete or accurate as the results of the original queries. A fundamental difference between the original and the simpler queries is that the simpler queries use only [Common Service Data Model \(CSDM\)](#) attributes for CI status (such as `life_cycle_stage`), while the default queries also use the legacy `operational_status` and `status` attributes. In an environment that hasn't migrated to CSDM, the simple queries yield fewer results.

You can examine (read access only) the simple query in the Simple Condition column in the `cmdb_group_contains_encoded_query` table.

To use the simpler set of queries, set the `sn_cmdb_ws.ci_overview.enable_simple_condition` system property to **true (false by default)**.

My CIs

Accessible to: CMDB Admin, CMDB Editor.

A chart of CIs managed by you or by the group assigned in the Managed by Group attribute and which you're a member of. CIs are grouped by up to 20 common class categories such as Applications, Devices, and Servers. If there are more than 20 classes to show, then all remaining classes are lumped into one additional bar on the chart.

Select a class bar to drill down into the CIs for the class. Then drill down to any [CI details pane](#) with a timeline, health overview, and several types of attributes such as key and discovery attributes.

Use the `sn_cmdb_ws.ci_overview.managed_by_me.enabled` property to show or hide this chart.

Cloud vs Non-cloud resources

Charts showing counts and details for resources that are hosted on various cloud services versus those resources that aren't, with breakdown by key CI classes such as applications, databases, and datacenters. Resources can be deployed on cloud services such as the Microsoft Azure Cloud, or on the local instance or other non-cloud solutions. For some ongoing operations in the organization, it might be necessary to have those details that can be difficult to obtain.

The following conditions must be met for the Cloud vs Non-cloud resources charts to appear and to show meaningful data:

- The table Cloud Service Accounts [cmdb_ci_cloud_service_account] must exist.
- The Logical Datacenter [cmdb_ci_logical_datacenter] table must contain at least one record for a cloud datacenter.
-

The Datacenter Types [sn_cmdb_ws_datacenter_type] table must contain at least one record for a datacenter that is classified as cloud storage in the organization. In the base system, this table is pre-populated with several records for common cloud services such as the Azure Datacenter [cmdb_ci_azure_datacenter] class. The chart calculates and shows data only for cloud services for which there's a record in the Datacenter Types [sn_cmdb_ws_datacenter_type] table.

For details about adding datacenters in your organization, with the cloud or non-cloud classification, see [Configure datacenters for the Cloud vs Non-cloud resources chart](#).

The following cloud vs Non-cloud charts are available:

- CI classes bar chart:

Each bar in the CI classes chart represents a pair of a CI class and a storage type (cloud, non-cloud), such as the bar for Applications/Cloud. For each bar, there's a scheduled job that runs every 24 hours to collect and calculate the data for the bar. The running time depends on the amount and complexity of the data that a job collects, which can be different for each scheduled job. Also, the schedules of the jobs are staggered so that they don't all run at the same time and

exhaust resources. For details about the class-specific criteria used for the chart calculations, see [Class-specific criteria for the Cloud vs Non-cloud resources chart](#).

If there's a CI class/type pair that isn't important in the organization, you can exclude that pair from the CI classes chart. For more details, see [Configure CI classes for the Cloud vs Non-cloud resources chart](#).

On the CI classes chart, you can:

- Select a bar to open the Cloud vs Non-cloud resources pane. Then, select either of the following tabs to drill down into further details for the bar:
 - **Cloud vs Non-cloud CIs**

Shows a bar chart for CIs stored on a cloud service and those CIs that aren't, per CI class. For each CI class, select the cloud or the non-cloud bar to show the CIs that are associated with the selected bar, in a list view. You can then select a CI from the list to further drill down to the CI details pane.

CI Classes by Cloud Providers

Pie charts per CI class with more granular details for those CIs that are hosted on cloud providers. Pie slices have randomly-selected colors, and they show a breakdown by cloud providers for various CI classes. For each pie, select a slice to show the CIs that are associated with the pie slice, in a list view. You can then select a CI from the list to further drill down to the CI details pane.

- Select **Latest updates** to see updated status for the scheduled jobs that produce the data for the chart. Status is color-coded to indicate whether the job has completed successfully (green), or failed to complete (red).
- Application Services pie chart:

The Application Services pie chart is available starting with CMDB Workspace v3.6.

The Application Services pie chart uses the Service Configuration Item Associations [svc_ci_assoc] table and checks the cloud/non-cloud status of application service CIs in the CI classes chart. Application services in the Application Services pie chart are classified as follows:

- Cloud: All of the CIs in the application service have been determined to be hosted on a cloud service.
- Non-cloud: All of the CIs in the application service have been determined not to be hosted on a cloud service.
- Hybrid: The application service contains a mixture of CIs where some are hosted and some aren't hosted on cloud services.
- Unknown: The cloud/non-cloud classification couldn't be determined because some of the application service CIs aren't classified as cloud or non-cloud CIs.

Note: Because classification of Application Services depends on the classification of CIs in the CI classes chart, there might be some discrepancy between the two charts. This can occur if an application service CI changed its cloud/non-cloud status, and the Application Services chart hasn't refreshed yet to reflect that change.

Select the Application Services chart to access the list views of the associated application services, grouped by cloud/non-cloud classifications. From those list views, you can drill down to the [Dependency Views](#) map for application services.

The scheduled jobs associated with the charts are set up with several hard-coded limits that if exceeded, result in failure conditions that are reflected in the job status. Timing out is set to two hours and the maximum number of records to collect is set to 500,000. If a job exceeds any of those limits, it's automatically stopped.

A scheduled job that can't complete for 3 (default) consecutive days is automatically disabled for future runs. You can manage the disabling of scheduled jobs, in the following ways:

- Modify the default number of consecutive days that are counted by adding the sn_cmdb_ws.insight.category.disable_after_failure system property to the System Properties [sys_properties] table and then setting its value. For more details, see [Add a system property](#).

- Resume a disabled scheduled job.

UI activity	Additional requirements
<ul style="list-style-type: none">• Mapped Application Service view• Application Service• Dynamic CI Group• Tag Based Service	app_service_user role

My work

Accessible to: CMDB Admin, CMDB Editor, CMDB User.

All open tasks from [CMDB Data Manager](#) that are assigned to you, or to the group assigned in the Managed by Group attribute and which you are a member of. Tasks can be, for example, attestation tasks and life cycle-related tasks. The list for CMDB Admins also includes such tasks that aren't assigned to anyone and CMDB Admins can then assign those unassigned tasks.

Select **Open tasks** and **Overdue tasks** to review and process the tasks.

CMDB Health

Accessible to: CMDB Admin, CMDB Editor, CMDB User.

Health metrics for CIs and relationships. Select the percentage numbers to navigate to the CMDB Health and CMDB Relationship Health dashboards:

- The Overall percentage number represents the health of all CIs as an aggregation of all three health Key Performance Indicators (KPIs). Those KPIs are correctness, compliance, and completeness, each consisting of sub-metrics.

- The Relationship percentage number represents the overall health of relationships as an aggregation of the orphan, duplicate, and stale relationships KPIs.

•

The color-coded status label follows the CMDB Health scorecard thresholds specified in the CI Class Manager. However, the labels 'Best', 'At Risk', and 'Critical' are replaced by 'Excellent', 'Fair', and 'Poor' respectively. For more information see, [Configure CMDB Health scorecard thresholds](#).

- Select **Factors impacting your score** to see the breakdown of the score percentage by the three key KPIs (Completeness, Compliance, and Correctness).

For more information, see [CMDB Health](#), [CMDB Health KPIs and metrics](#), and [View CI relationships health](#).

UI activity	Additional requirements
CMDB Health	asset or itil role Setup and configure CMDB Health

Total CIs

Accessible to: CMDB Admin, CMDB Editor, CMDB User.

Count and trend of the total number of CIs in the CMDB for the past 7 days.

Use the `sn_cmdb_ws.total_cis.enabled` property to show or hide this chart.

Quick links

A list of links to key CMDB dashboards and tools. You can [add a link](#) to the list of quick links that are available to you.

- **Dependency View:** Provides a graphic infrastructure view for a CI and any application or business services that it's part of and that it supports.
- **Query Builder:** Easily build complex infrastructure and service queries, that span multiple CMDB classes, non-CMDB tables, and that can involve many CIs that are connected by different relationships.
- **CMDB Data Manager:** Centrally create, edit, review, publish, and track Data Manager policies and the tasks generated by the policies.
- **CI Class List:** List view of CMDB CIs grouped by common classes.
- **Unified Map:** Graphical map showing the hierarchy of CIs and the relationships between them, and application services. Directly from the map, you can access attributes of CIs and relationships, and related items such as changes, incidents, and problems for a CI.

UI activity	Accessibility	Additional requirements
Dependency View	CMDB Admin, CMDB Editor, CMDB User	dependency_views role
Query Builder	CMDB Admin, CMDB Editor, CMDB User	cmdb_query_builder_read role
Data Manager	CMDB Admin	
Add or edit a custom quick link	CMDB Admin, CMDB Editor, CMDB User	

Shared pages

For information about the shared pages, see the Dev site as follows:

- [CI Service Relationships](#)

- CI Infrastructure Relationships

CI details pane

When you drill down to a CI record, the following details for the CI appear:

- CI Timeline - Last 14 days: A timeline of CI activities such as change requests.
-

CI Health: A summary of the health of the CI, showing related items such as critical incidents, incomplete attributes, and stale relationships for the CI.

Role requirement: itil (for accessing incidents).

- Details: CI attributes, grouped into categories such as Key attributes, Asset attributes, Discovery attributes, Operational attributes, and More attributes.

Note: Use the CMDB - Workspace form view for a CI class to configure which attributes appear.

- Activity: An activity stream to track what's changed in the CI record.
- Infrastructure Relationships: List of the infrastructure CIs related to the CI.
- Service Relationships: List of business applications, service offerings, and application services that the CI may be related to.

On the CI details pane, you can:

- Select **Open Dependency View** to open the **Dependency Views** map and display a graphic infrastructure view of the specific CI record.
- Select **View CMDB 360 Data** to show CMDB 360 details at the CI attribute level for the specific CI record.
-

Select **Save** to save any changes made to attributes for the CI record.

- Select the More Actions icon (...) for additional functions:

- Select **Create Change** to [create a new change request](#) for the CI record.
- Select **Create Incident** to [create a new incident](#) for the CI record.
- Select **Delete** to delete the CI record.

UI activity	Additional requirements
CI Details Accessible to: CMDB Admin, CMDB Editor, CMDB User	
CI Health	itil
Related Open Changes Accessible to: CMDB Admin, CMDB Editor, CMDB User	sn_change_read role
Related Incidents Accessible to: CMDB Admin, CMDB Editor, CMDB User	sn_incident_read role
Related Alerts Accessible to: CMDB Admin, CMDB Editor, CMDB User	Event Management (com.glideapp.itom.snac) plugin evt_mgmt_user role Set up Event Management

UI activity	Additional requirements
Related Application Services Accessible to: CMDB Admin, CMDB Editor, CMDB User	app_service_user role
View CMDB 360 Data Accessible to: CMDB Admin, CMDB Editor, CMDB User	Enable and configure CMDB 360
Save Accessible to: CMDB Admin	
More Actions/Delete Accessible to: CMDB Admin	

Governance view in CMDB Workspace

Use the Governance view in CMDB Workspace to manage your tasks. Tasks in this view are related to data compliance such as attestation tasks.

Access

Role requirement: sn_cmdb_admin (CMDB Admin), sn_cmdb_editor (CMDB Editor), or sn_cmdb_user (CMDB User).

To access the Governance view, navigate to **Workspaces > CMDB Workspace** and then select **Governance** in the CMDB Workspace menu bar.

Governing data

The My Work tile lists any attestation tasks assigned to you or to an assignment group that you belong to in accordance with [CMDB Data Manager](#) Attestation policies. Review and process these attestation tasks by checking the physical existence of IT infrastructure or applications associated with the CIs in the tasks. For information about reviewing and processing attestation tasks, see [Review CMDB Data Manager Attestation tasks](#).

The **Overdue tasks** tab lists those tasks that are overdue so you can review those tasks at a higher priority.

CMDB 360 view in CMDB Workspace

The CMDB 360 dashboard provides aggregations and analysis of CMDB 360 data. CMDB 360 collects data about all the discovery sources reporting attribute values for CIs. Use the CMDB 360 view in Configuration Management Database (CMDB) Workspace to track activities and identify potential issues of discovery sources. You can also create your own queries and associated schedules and reports to explore CMDB 360 data.

- For concepts and other background information about CMDB 360/Multisource CMDB, see [CMDB 360/Multisource CMDB](#).
- For information about all CMDB 360 dashboard settings, see [Configure the CMDB 360 dashboard](#).
- For information about using the CMDB 360 view, see [CMDB 360 experience in CMDB Workspace](#).

Note: Most cards on the CMDB 360 dashboard support non-CMDB tables in their aggregation, or can be configured to provide support. However, the CIs not reported by discovery sources card, for example, doesn't apply to non-CMDB tables. Creating queries for non-CMDB tables is also supported. For information about support for non-CMDB tables, see [IRE support for non-CMDB tables](#).

Access

Requirements:

- Role requirement: sn_cmdb_user (CMDB user) or any role containing sn_cmdb_user
- Additional requirement: [Enable and configure CMDB 360](#)

To access the CMDB 360 view in the CMDB Workspace, navigate to **Workspaces > CMDB Workspace**. In the CMDB Workspace menu bar, select **CMDB 360**.

Management view in CMDB Workspace

Overview with deeper insights into CMDB health and activities that CMDB Admins can use for management. This view provides access to key management tools and details such as recent activities in the CMDB and duplicate CIs.

Access

Role requirement: sn_cmdb_admin (CMDB Admin)

To access the Management view, navigate to **Workspaces > CMDB Workspace** and then select **Management** in the CMDB Workspace menu bar.

CMDB Data Manager

The following cards show details about CIs and policies associated with [CMDB Data Manager](#):

Rejected CIs

CIs set as rejected during a review of [attestation tasks](#).

Excluded CIs

CIs set as [excluded](#) from CMDB Data Manager policies.

Draft policies

Draft [Data Manager policies](#) that were created but not published.

CI Correctness

Cards in this section show health state for the sub metrics of the [CMDB Health](#) correctness KPI. Counts are based on testing CIs against pre-defined data integrity rules such as:

- [Identification rules](#) (to detect duplicate CIs)
- [Orphan CI rules](#)
- [Staleness CI rules](#)

You can set the time interval used in calculations for these counts.

For more information about the correctness KPI, and the duplicate, orphan, and stale sub metrics, see [CMDB Health KPIs and metrics](#).

Recent activity trends

The following cards show recent activities in the CMDB:

Recent CI Activity

A 7-day chart showing metrics related to CIs such as the number of new CIs and updated CIs.

Recent Application Service Activity

A 7-day chart showing metrics related to [Application Services](#) such as the total number of Application Services, new and updated Application Services, and the number of Application Services with outages.

Management Tools

Provides links that you can use to access CMDB dashboards, tools, and list views. The links are grouped by categories as described below.

Note: Some links are conditionally available based on installation of applications, active plugins, and your assigned role. For a link that doesn't appear, make sure that all the requirements for the link are met.

Manage:

-

CI Class Manager: Centrally view, create, or edit class definitions and class settings for Identification and Reconciliation (IRE) and for CMDB Health.

Additional role requirement: itil or personalize_dictionary

- **CMDB Data Manager:** Centrally create, edit, review, publish, and track Data Manager policies and the tasks generated by the policies.
- **CMDB groups:** Show a list view of current CMDB groups where you can create new CMDB groups, and manage the existing CMDB groups. A CMDB group is a collection of CIs to which you can apply CI actions collectively to all the CIs in the group. For example, CMDB Health can monitor CIs in a Health-type CMDB group, and report the aggregated health for the group as a whole.
- **Dynamic CI Group:** Show a list view of current dynamic CI groups where you can create new dynamic CI groups, and manage the existing ones. Dynamic CI Groups act as application services that are populated with members of the CMDB group that is associated with the dynamic CI group. For more information about dynamic CI groups, see [Application services](#).

Optimize:

- **CMDB Health:** View the CMDB Health dashboards and configure the CMDB health KPIs and metrics that CIs are evaluated by in CMDB Health dashboards.

Additional requirements:

- Set up and configure CMDB Health
 - Roles: asset or itil
- **CMDB Data Foundations Dashboard:** Displays the CMDB Data Foundations dashboard.
- Additional requirements:
- Set up CSDM and CMDB Data Foundations Dashboards
 - Roles: asset, Itil_admin, or admin
 - Plugin: com.snc.cmdb.getwell

- **De-Duplication Tasks:** Remediate a de-duplication task by using the Duplicate CI Remediator wizard which guides you through the duplicate CI reconciliation process.

Additional role requirement: itil

- **CMDB Remediation Rules:** Rules associated with a [CMDB Health](#) task that was created for a failed CMDB Health test. A CMDB remediation rule runs a remediation workflow to remediate an issue reported by CMDB Health.

Visualize:

- **Dependency Views:** Provides a graphic infrastructure view for a CI and any applications or business services that it is part of and that it supports.

Additional role requirement: dependency_views

- **Query Builder:** Easily build complex infrastructure and service queries, that span multiple CMDB classes, non-CMDB tables, and that can involve many CIs that are connected by different relationships.

Additional role requirement: cmdb_query_builder_read

Create: [New Technical Service](#)

Additional role requirement: service_admin

Insights view in CMDB Workspace

Use the Insights view in CMDB Workspace to see and increase the level of adoption of key CMDB features and application services to improve the overall health of the CMDB. Explore how tools and features can maximize the health and efficiency of your CMDB and use direct links to install and start using features immediately.

Insights view is available starting with CMDB Workspace v3.4.6.

Access

Role requirement: sn_cmdb_user (CMDB User), or a user role containing sn_cmdb_user (sn_cmdb_admin, sn_cmdb_editor).

To access the Insights view, navigate to **Workspaces > CMDB Workspace** and then select **Insights** in the CMDB Workspace menu bar.

The Insights view includes the following tiles:

- CMDB feature adoption
- CMDB performance insights
- Application services
- Cloud vs Non-cloud resources

CMDB feature adoption

Shows dial charts for the overall adoption levels of all three categories of CMDB tools and features:

Data ingestion

Tools and features that support and ensure ingestion of high-quality data into the CMDB.

Data governance

Tools and features that let you manage the CMDB data after it has been ingested.

Search and analytics

Tools and features that provide meaningful and helpful insights into the CMDB data.

Select each dial chart in the tile to access its tab and associated cards with further details about the features in the category. Pay special attention to categories with low levels of adoption and those features that aren't yet implemented in the instance and which you should consider for adoption.

Each category tab shows cards for the features in the category and the following general cards:

- Overall adoption level for the tab: The calculated overall level of adoption for the tab. Calculations are specific for each tab, and can be based, for example, on the level of adoption of some or all the features in the tab.

- Adoption progress: Installation, activation, or usage status per feature in the tab. Select a feature link to access a relevant resource such as the feature's landing page with an overview dashboard or an installation location.

Depending on adoption level, a card might contain any of the following resource links:

- Learn more:** Link to documentation to learn and explore the benefits and usage of the feature.
- View demo:** Link to a short demo about the feature.
- Get started:** Link to a landing page where you can start immediately utilizing the feature.
- Install app:** Link to the ServiceNow Store where you can immediately install the app.

Cards use different methods to examine the instance and determine if the card's feature is installed, activated, and being used. For example, some cards check for the installation status of plugins and some cards rely on data in the specific feature tables. Many counts and aggregation data that appear on cards is based on [Performance Analytics indicators](#) built on top of the Base Aggregate Data [sn_cmdb_ws_base_aggregate_data] table.

- If the result is that the card's feature is installed or being used, then the card shows charts and counts about the level of usage. In which case, the card's label might slightly adjust.
- If the result is that the card's feature isn't installed or isn't in use, then links are provided to resources where you can explore, install, and start using the feature.

Most of the cards scheduled jobs run every 24 hours, therefore, depending on the type of data, some card data is based on recent but not current data. The Last updated timestamp in the cards reflects the collection time for the data that was used for the card. Also, immediately after getting started with a feature, a card won't reflect on the latest status or usage of the feature, until up to 24 hours when the card's scheduled job runs (for the CIs processed by IRE based on source card, the associated scheduled job, CMDB Workspace - Populate aggregates Monthly, runs monthly).

The following sections provide details for each card, including the calculation script used for the card.

CMDB feature adoption: Data ingestion

The overall adoption level for data ingestion maps to the following findings in the instance:

- Low (red): Less than 80% of CIs are processed by Identification and Reconciliation Engine (IRE)
- Moderate (amber): 80—90% of CIs are processed by IRE
- High (green): Over 90% of CIs are processed by IRE

Data ingestion contains the following features and aggregations:

CIs processed by IRE

Determines the percentage of CIs that are processed by [IRE](#), by checking the Source [sys_object_source] table. CIs that aren't processed by IRE introduce a data integrity risk.

The percentage of CIs that are being processed by IRE determines both, the level of adoption for this feature card and the overall adoption level for the entire data ingestion category.

Service Graph Connectors

Determines the installation and usage status of [Service Graph Connectors](#), by checking:

- If the ITOM Licensing plugin (com.snc.itom.license) is active
- If there is at least one Service Graph Connector installed in the instance

If Service Graph Connectors are installed and are in use, then the card shows a count of those connectors.

Note: The count of Service Graph connectors that appears on the card might be slightly different than the number of connectors that show on the ServiceNow Store site because the counting methods that are used are different.

IntegrationHub ETL

Determines whether the [IntegrationHub ETL](#) store app is installed and used, based on records in the CMDB Integration Studio Application Data Sources [cmdb_inst_application_feed] table. If IntegrationHub ETL is in use, then the card shows a count of ETL transform maps in IntegrationHub ETL (demo ETL transform maps aren't counted).

Select [View ETL transform maps](#) to open IntegrationHub ETL where you can examine existing ETL transform maps and create new ones.

CIs processed by IRE based on source

Chart showing CIs processed by IRE, grouped by Service Graph Connectors, ServiceNow Discovery, a combination of both, and others, for the past six months.

The following discovery sources are counted as ServiceNow Discovery:

- ServiceNow
- ServiceWatch
- ACC-Visibility
- AgentClientCollector
- CredentiallessDiscovery

Point to the chart to show monthly aggregation data.

This card is hidden if there are no CIs processed by IRE.

CMDB application for APIs and CLI

Determines whether the [CMDB application for APIs and CLI](#) store app is installed.

This card appears only if the store app isn't installed, providing helpful resources for exploration and adoption.

CMDB feature adoption: Data governance

The overall adoption level for data governance maps to the following findings in the instance:

- Low (red): Less than 33% of features are used
- Moderate (amber): 33—66% of features are used
- High (green): Over 66% of features are used

The overall level of adoption of data governance is based on whether [CMDB Data Manager](#) features are used.

Note: Historical data might not be available for all past 90 days because CMDB Workspace version 3.4, which relies on Performance Analytics indicators to collect and save historical usage data, was deployed less than 90 days ago. This situation might result in a discrepancy between actual usage and what the card shows.

CMDB Data Manager/CIs used in Data Manager policies

Determines usage by checking if either of the following conditions is true:

- There are any user-created Delete, Retire, or Archive policies (by searching table CMDB Data Manager Policy and Attributes [cmdb_data_manager_policy_and_attributes])
- There are any CIs processed by these user-created Delete, Retire, or Archive policies, in the last 90 days (by searching table CMDB Data Management Policy Executions [cmdb_data_management_policy_execution])

If CMDB Data Manager is in use, shows a chart with CIs that were processed by these policies in the past six months, by month, and by policy type.

Data attestation/CIs used in data attestation

Determines usage by checking if either of the following conditions is true:

- There are any user-created [Attestation policies](#) (by searching table CMDB Data Manager Policy and Attributes [cmdb_data_manager_policy_and_attributes])
- There are any CIs processed by these user-created Attestation policies, in the last 90 days (by searching table CMDB Data Management Policy Executions [cmdb_data_management_policy_execution])

If data attestation is in use, shows a chart with CIs that were processed by attestation policy tasks in the past six months, by month.

Data synchronization

Checks if there is at least one class for which the managed_by_group attribute is globally set so that all class CIs are synchronized on the same value.

For information about synchronizing group assignment attributes using the CI Class Manager, see [Set the group for a CI or an entire class of CIs](#).

The Data synchronization card is available starting with CMDB Workspace v3.6.

Principal classes

Checks if the Principal Class filter is configured with at least one principal class. The Principal Class filter limits the number of CIs that appear in list views, to show only CIs of principal classes. Reducing the amount of data in list views to only relevant data, improves performance and efficiency.

For more information about managing the Principal Class filter in CI Class Manager, see [Update the list of classes in the Principal Class filter](#).

The Principal class card is available starting with CMDB Workspace v3.6.

CMDB feature adoption: Search & analytics

The overall adoption level for search & analytics maps to the following findings in the instance:

- Low (red): Less than 33% of features are used
- Moderate (amber): 33—66% of features are used
- High (green): Over 66% of features are used

Search & analytics contains the following features and aggregations:

CMDB Query Builder/Query Builder queries

Determines if the [CMDB Query Builder](#) is in use, by checking if either of the following conditions is true:

- There are any records in the Saved Queries [qb_saved_query] table in which Source is **QB**
- There are any queries executed or queries executed with reports, in the last 90 days

Note: Historical data might not be available for all past 90 days because CMDB Workspace version 3.4, which relies on Performance Analytics indicators to collect and save historical usage data, was deployed less than 90 days ago. This situation might result in a discrepancy between actual usage and what the card shows.

If CMDB Query Builder is in use, shows a chart with counts of query executions and query executions in reports, for the past six months, by month.

Point to the chart to see monthly aggregation data.

Intelligent search

Determines if Intelligent Search for CMDB is in use by checking the NLQ Query Logs [nlq_query_log] table for any records where source is **CMDB_WS**, from the past 90 days. If Intelligent Search is in use, shows counts of Intelligent Search queries for the past six months, by month.

Note: Historical data might not be available for all past 90 days because CMDB Workspace version 3.4, which relies on Performance Analytics indicators to collect and save historical usage data, was deployed less than 90 days ago. This situation might result in a discrepancy between actual usage and what the card shows.

CMDB 360 — Records in Multisource

Total number of raw [CMDB 360](#) records in the CMDB 360 data store that contains records for each discovery source report, per each CI attribute. This card is identical to the [Total CMDB 360 records](#) card in the Discovery Sources tile in the CMDB 360 view.

This card appears only if CMDB 360 is enabled, which is determined by checking the ITOM Discovery License (com.snc.itom.discovery.license) plugin and the system property [glide.identification_engine.multisource_enabled](#).

CMDB 360 queries

Count of CMDB 360 queries that exist in the CMDB Multisource Queries [cmdb_multisource_query] table.

This card appears only if CMDB 360 is enabled, which is determined by checking the ITOM Discovery License (com.snc.itom.discovery.license) plugin and the system property `glide.identification_engine.multisource_enabled`.

CMDB Data Foundation dashboard

Determines if the [CMDB and CSDM Data Foundations Dashboards](#) store app (which includes the CMDB Data Foundation dashboard) is installed.

This card appears only if the store app isn't installed, providing helpful resources for exploration and adoption. However, the feature is still included in calculating the search & analytics overall level of adoption.

CMDB Health Dashboard

Determines if [CMDB Health](#) is in use by checking if at least one CMDB Health Dashboard job is enabled.

This card appears only if the feature isn't in use, providing helpful resources for exploration and adoption. However, the feature is still included in calculating the search & analytics overall level of adoption.

CMDB performance insights

CMDB performance insights is available starting with CMDB Workspace v3.6 and only appears for users with the sn_cmdb_admin (CMDB Admin) role.

CMDB performance insights helps you understand the ways in which your configurations impact the performance of your CMDB. You can use the charts and tools within CMDB performance insights to troubleshoot, debug, or diagnose performance issues. CMDB performance insights also analyzes your CMDB and Service Graph Connectors on your instance to generate recommendations on how you can improve the performance of your CMDB. The CMDB performance insights tile itself can show up to two of those recommendations.

Select **View performance insights** to access the CMDB performance insights data.

The Payloads & CIs tab contains the following tiles:

Partial payloads

Partial payloads occur when the data source didn't provide enough information to uniquely identify the CI, preventing IRE from processing the CI.

- Total partial payload count:

Shows the total number of partial payloads that exist in your instance. Large numbers of partial payloads in your instance can cause performance deterioration of the CMDB.

For more information about partial payloads, see [Identification and Reconciliation engine \(IRE\)](#).

- Discovery source:

Breaks down the number of partial payloads by discovery sources. You can drill down on slices in this pie chart to see the list of partial payloads with errors, filtered by discovery source.

You can drill down on specific partial payloads to better understand the error that you're experiencing. When you drill down on a partial payload, you can see the full payload item, which you can review to troubleshoot and address the specific errors.

For more information about error types, see [Generate and simulate payload execution using identification simulation](#).

Related records

Data about related records that are missing references. A record is missing a reference when the Referenced field for that record is empty.

- Related records missing reference:

Shows the total number of records that do not reference a CI in the Referenced field in the Related table. You can see a full list of the Related tables and the associated Referenced fields in the Related Entries [cmdb_related_entry] table.

For more information about Related tables, see [Configuration Management and the CMDB](#).

- Related records missing reference by table:

Breaks down the number of CMDB records missing references, by table. You can drill down on slices in this chart to see the list of specific records filtered by table.

Before you can drill down on a slice of this pie chart, you must have any user roles required to view the table.

You can also select **New** from the list view to create a new related entry table record. For more information about creating or editing a related entry table record, see [Edit a related table from CMDB performance insights](#).

Duplicate and stale CIs

A CI is flagged as duplicate during identification and reconciliation. A CI is flagged as stale if it has not been updated within the Effective Duration time period specified in the [CMDB Health staleness rule](#) for the CI class.

- CIs:

Shows the number of CIs that are either duplicate or stale.

- Stale records by class:

Breaks down the number of stale CIs based on the CI class. You can drill down on slices in this pie chart to see the list of specific CI records filtered by class.

For more information about stale CIs, see [CMDB Health KPIs and metrics](#).

- Duplicate records by class:

Breaks down the number of duplicate CIs based on the CI class. You can drill down on slices in this pie chart to see the list of specific CI records.

For more information about duplicate CIs, see [Duplicate CIs](#).

Relationship records missing parent or child

Shows the trend line and number of relationship records in the CI Relationships [cmdb_rel_ci] table that are missing a parent or child CI. CIs that are missing a parent or child are considered invalid records and can have a performance impact on your instance.

Recommendations

Recommendations on this panel include a link to the related documentation and typically enable direct access to the associated tool, feature, or system property. Expand or minimize the panel by selecting the light bulb icon.

The first two recommendations appear on the CMDB performance insights tile of the Insights view.

You can only see a recommendation if you have the roles needed to access the feature, tool, or system property. Users with the sn_cmdb_admin role can hide recommendations or adjust the order in which recommendations appear. Users can use the **Active** and **Order** fields in the CMDB WS Imp Action Card Config [sn_cmdb_ws_imp_action_card_config] table to configure those elements.

The Service Graph connectors tab contains the following tiles:

Note:

- The Service Graph connectors tab only appears if your instance has at least one Service Graph Connector and you have the cmdb_inst_admin role.
- To edit data source or scheduled data import records from CMDB Workspace, you may need to set **Application scope** to the **Application** of the data source or data import.

Connectors data source

Configurations on your Service Graph Connector data sources affect the ingestion and processing of incoming data. Changing your Service Graph Connector data source configurations can streamline data

handling, making your CMDB more efficient and reducing impacts on the performance of your instance.

- Sources with batch processing turned off:

Shows the percentage of Service Graph Connector data sources in the Data Sources [sys_data_source] table with disabled **Use Batch Import**. It also lists the total number of data sources, and the number of data sources with enabled or disabled batch processing.

To enable batch processing, access the record of a specific data source from the list view and select **Use Batch Import**. For more information about batch processing, see [Data source fields](#).

You can also select **Edit** to update a data source from the list view. For more information about editing a data source, see [Edit a data source from CMDB performance insights](#).

- Sources with concurrent import turned off:

Shows the percentage of scheduled data imports of Service Graph Connector data sources in the Scheduled Data Imports [scheduled_import_set] table with **Concurrent Import** turned off. It also lists the total number of scheduled data imports, and the number of scheduled data imports with **Concurrent Import** turned on or off.

To enable concurrent import, access the record of a specific scheduled data import from the list view and select **Concurrent Import**. For more information about concurrent import, see [Concurrent imports](#).

You can also select **Edit** to update a scheduled data import from the list view. For more information about editing a scheduled data import, see [Edit a scheduled data import from CMDB performance insights](#).

- Sources with non-custom size partition method:

Shows the percentage of scheduled data imports of Service Graph Connector data sources in the Scheduled Data Imports [scheduled_import_set] table that use a non-custom size partition method. It also lists the total number of scheduled data imports, and the number of scheduled data imports that use a non-custom size partition method.

To use a custom size partition method, access the record of a specific scheduled data import from the list view. Ensure that **Concurrent Import** is selected. From the **Partition Method** drop-down menu that appears, select **Custom size**. For more information about partition methods, see [Schedule a data import](#).

You can also select **Edit** to update a new scheduled data import from the list view. For more information about editing a scheduled data import, see [Edit a scheduled data import from CMDB performance insights](#).

Connectors execution trends

Aggregates Service Graph Connectors with outlier connector executions, where the number of imported rows or the rate of processing is significantly higher or lower than the 30-day trend.

- Connectors with processing rate outliers:

Shows the number of Service Graph Connectors that are considered outliers with connector execution processing rates outside the expected trend lines within the past 30 days.

- Connectors with import count outliers:

Shows the number of Service Graph Connectors that are considered outliers with connector execution import counts outside the expected trend lines within the past 30 days.

You can drill down the cards in the Connectors execution trends tile to access the Service Graph connector execution trends window. In that window, a Service Graph Connector is available in the connectors drop-down menu only if it has an execution record.

Click the tabs on the Service Graph connector execution trends window to view the following charts and the Connector Execution list view of executed connector import sets:

- Processing rate:

Models the trend line of processing rates for connector executions. Shows the rate at which a connector processes rows of data over a period of time.

There are two zones, which are the Confidence Band and the Prediction Band. Service Graph Connectors with connection executions that have processing rates outside of the Prediction Band are considered outliers. These zones don't appear when you select more than one Service Graph Connector.

- Import count:

Models the trend line of rows processed for connector executions. Shows the number of rows a connector processes over a period of time.

There are two zones, which are the Confidence Band and the Prediction Band. Service Graph Connectors with connector executions that have import row counts outside of the Prediction Band are considered outliers.

On both charts, you can select outlier and non-outlier Service Graph Connectors to see the trend lines against each other.

For more information about processing data with Service Graph Connectors, see [Service Graph Connectors](#).

Application services (Application Services Dashboard)

Shows a chart with a count of [application services](#) in your organization, based on records in the Application Services [cmdb_ci_service_auto] table. The chart shows the trend of total number of application services per day, for the past seven days. Use the Application Services Dashboard to monitor the adoption level and health of application services.

Note: Some population methods for application services are available only with [Service Mapping](#) and therefore won't appear in the tile if Service Mapping isn't installed.

Select the chart to access the tile tabs and their cards with further details. Pay special attention to application services that aren't fully configured and are missing data. Reduce the number of incomplete application services by [editing application services](#) and populating any empty attributes.

Select any card to drill down to the list view of the respective application services.

The Overview tab shows the following aggregations:

- Total application services: Count of all application services.
- Population method defined: Count of application services for which a population method is specified.
- Population method not defined: Count of application services for which a population method isn't specified.
- Application services types: Chart of application services by population methods such as Dynamic CI Group and Manual, including application services without a population method (Empty). The chart includes business services that were converted to application services.
- Application services missing data: Chart of application services by key data, such as service offering and owner, that is missing.

The Application service coverage tab contains the following tiles and cards:

- Application Servers
 - Total servers: Total number of application servers.
 - Servers not in application service: Count of application servers which aren't in any application service.
 - Servers not in an application service: Chart of application servers which aren't in any application service, by class.
- Databases
 - Total databases: Total number of databases
 - Databases not in application service: Count of databases which aren't in any application service.
 - Databases not in an application service: Chart of databases which aren't in any application service, by class.
- Hardware Servers
 - Total hardware servers: Total number of hardware servers.

- Hardware servers not in application service: Count of hardware servers which aren't in any application service.
- Hardware servers not in an application service: Chart of hardware servers which aren't in any application service, by class.

Note: Application servers, hardware servers, and databases not included in application services, are counted only up to about 100,000, even if the actual number is over this limit. This number limit is set by the `glide.cmdb.csdm.app_service.max_results` property.

Select **Owned by**, **Support group**, or **Change group** to filter the list of application services that are included in the dashboard, by a key attribute. A filter doesn't impact counts of application services in which the respective filter attribute is missing. For example, filtering by Owner, doesn't change the count of the Missing Owner card.

Cloud vs Non-cloud resources

Charts showing counts and details for resources and application services that are hosted on various cloud services versus those that aren't.

Resources and application services can be deployed on cloud services such as the Microsoft Azure Cloud, or on the local instance or other non-cloud solutions. For some ongoing operations in the organization, it might be necessary to have those details which can be difficult to obtain.

For details about the charts, see 'Cloud vs Non-cloud resources' in the [CMDB Workspace store app](#) topic.

The Application Services pie chart is available starting with CMDB Workspace v3.6.

Modify important actions in CMDB Workspace

Modify the appearance order and other properties of the important action cards that appear on the landing page of the CMDB Workspace.

Before you begin

Role required: sn_cmdb_admin

About this task

The task cards that appear in the Important tasks tile on the landing page of the CMDB Workspace, are stored in the CMDB WS Imp Action Card Config [sn_cmdb_ws_imp_action_card_config] table. You can edit some attributes of these cards, including the setting that controls whether a specific card appears at all. Cards in the Important actions tile are available only for the CMDB Admin [sn_cmdb_admin] and the CMDB Editor [sn_cmdb_editor] user roles and each is configured per one of those roles.

There is some overlap in the task cards that appear for a CMDB Admin and for a CMDB Editor, however, the filters in those task cards are different. Typically, for CMDB Admins, the card filter also includes unassigned tasks that a CMDB Admin needs to assign.

Only editing is possible, you can't add or delete action records in this table.

Procedure

1. Navigate to **All**.
2. In the navigation filter, enter `sn_cmdb_ws_imp_action_card_config.list`.
3. In the list view of the table, edit the table row that you want to modify or click the row to open the record form.

Field	Description
Persona	<p>The logged on user (such as CMDB Admin) for which this card appears.</p> <p>You can't edit this field.</p>
Type	Tasks group that this task belongs to, such as Duplicate CIs.

Field	Description
	You can't edit this field.
Table	The table used in the card filter. You can't edit this field.
Active	Determines whether the card for this action appears in CMDB Workspace.
Order	Numeric value that determines the order of each card within the rest of the cards in the Important actions tile. Cards with lower order numbers appear before cards with higher order numbers.
Filter condition	Filters for the tasks that are included for the card.
List columns	Columns that appear in the list view when you click a card to show its associated tasks.
List groupby	Attribute to group by the card's associated tasks in the card's list view. The card's list view appears when you click a card to show its associated tasks.

4. Click **Update**.

Add a quick link to the CMDB Workspace

For an immediate access to tools or data that you need, add your own quick link to the landing page of the CMDB Workspace.

Before you begin

Role required: sn_cmdb_admin, sn_cmdb_editor, or sn_cmdb_user

About this task

Quick links are stored in the Quick Links [sn_cmdb_ws_quick_links] table including those that are included in the base system. Only the user that added a quick link has access to that link, and can subsequently edit or delete that added link.

Procedure

1. Navigate to **Workspaces > CMDB Workspace**.
2. In the Quick links section, click the '+' icon.
3. Enter the **URL** and **Display text** for the link and then click **Add**.

A valid URL must start with "http:" or "https:". To link to a tool in the instance such as the CI Class Manager, enter the full URL to the tool's landing page.

4. Click **Done**.

Result

The new link is available only for the user that created the link, in the Quick links section in the landing page of the CMDB Workspace.

What to do next

Click on the Edit quick links icon to edit, delete, or reposition an added link. Then use the up and down arrows to move a link within the list, and click the Edit icon next to a link that you want to modify or delete from the list.

Configure datacenters for the Cloud vs Non-cloud resources chart

The Cloud vs Non-cloud resources chart in CMDB Workspace determines which Cls and application services are stored on a cloud and which

aren't in the organization. In addition to base system classifications, you can add cloud versus non-cloud classifications that reflect specific datacenter deployments in the organization.

Before you begin

Role required: sn_cmdb_admin

About this task

The Datacenter Types [sn_cmdb_ws_datacenter_type] table stores datacenter classes with a classification of being used as a cloud storage or not in the organization. The calculations for the Cloud vs Non-cloud resources chart in the CI overview tile in [CMDB Workspace](#), reflects on the datacenters in that table and their classifications.

ClIs are counted as non-cloud mainly if either of the following conditions is met:

- The CI is not associated with any datacenter.
- The CI is associated with a datacenter that is classified as non-cloud in the Datacenter Types table.

In the base system, the Datacenter Types table contains several common datacenters. For example, the Azure Datacenter [cmdb_ci_azure_datacenter] class is classified as 'Cloud'. Therefore, the Cloud vs Non-cloud resources chart includes in its calculations and bars the Azure Datacenter [cmdb_ci_azure_datacenter] class.

Procedure

1. Select **All**.
2. In the Filter navigator, enter `sn_cmdb_ws_datacenter_type.list` to access the Datacenter Types table.
3. Click **New** and fill out the Datacenter Type form for a datacenter that is used in the organization.

Field	Description
Datacenter class	A child class of the Logical Datacenter [cmdb_ci_logical_datacenter] class which is used in the organization.
Type	Classification that reflects whether the specified Datacenter class is being used as cloud storage or not.
Cloud Provider	Custom label for the cloud provider.

4. Click **Submit**.

Resume a disabled scheduled job for the Cloud vs Non-cloud resources chart

Resume a scheduled job that has been disabled, so that it resumes data collection for the Cloud vs Non-cloud resources chart in CMDB Workspace.

Before you begin

Role required: cmdb_query_builder (contained in the sn_cmdb_user, sn_cmdb_editor, sn_cmdb_admin user roles)

About this task

The [Cloud vs Non-cloud resources](#) chart in CMDB Workspace uses several scheduled jobs that gather and calculate the data for the chart bars and pies. Each bar in the CI classes chart represents a pair of a CI class and a storage type (cloud, non-cloud) such as Applications/Cloud. Each bar is associated with its own scheduled job. When a scheduled job exceeds its limits of time and amount of collected data, it is automatically stopped. A job that can't complete for 3 (default) consecutive days is automatically disabled for future runs.

Use the following procedure to later resume that disabled scheduled job.

Procedure

1. Click **All**.
2. In the Filter navigator, enter `sysauto_query_builder.list` to open the Scheduled Email of Query Builders table.
3. In the Scheduled Email of Query Builders list view, set Active to **true** for the scheduled job you want to resume.

Configure CI classes for the Cloud vs Non-cloud resources chart

Include or exclude pairs of CI Class/Type in the Cloud vs Non-cloud resources chart in CMDB Workspace.

Before you begin

Role required: `sn_cmdb_admin`

About this task

The CMDB Insight Query Categories [`sn_cmdb_ws_insight_query_category`] table contains the pairs of CI Class/Type for the Cloud vs Non-cloud resources chart in the CI overview tile in [CMDB Workspace](#). The Active setting in a record determines if the respective CI Class/Type pair appears in the chart. By default, all pairs are configured to appear in the chart.

A CI Class/Type pair appears or doesn't appear according to its Active setting and regardless of the status of its associated scheduled job.

Procedure

1. Select **All**.
2. In the Filter navigator, enter `sn_cmdb_ws_insight_query_category.list` to access the CMDB Insight Query Categories table.

3. Set Active to **true** or **false** for a CI Class/Type pair.

For example, set Active to **false** for the **Applications/Non-cloud** pair. This setting will exclude from the chart all CIs in the Applications class which are determined to be non-cloud.

Class-specific criteria for the Cloud vs Non-cloud resources chart

The Cloud vs Non-cloud resources chart provides counts for several key classes. The chart uses different classes and relationship criteria for each class to determine which resources count as cloud and which count as non-cloud.

The Cloud vs Non-cloud resources chart shows in the CI overview tile in the [CMDB Workspace store app](#).

Virtual Machine Instance [cmdb_ci_vm_instance]:

- Cloud:

Virtual Machine Instance [cmdb_ci_vm_instance] -> (Hosted on::Hosts) -> Datacenter class (one of the [configured cloud datacenters](#)) -> (Hosted on::Hosts) -> Cloud Service Account [cmdb_ci_cloud_service-account]

- Non-Cloud:

Virtual Machine Instance [cmdb_ci_vm_instance] -> (Hosted on::Hosts) -> Datacenter class (one of the [configured non-cloud datacenters](#))

- Total: Equals the record count in the Virtual Machine Instance [cmdb_ci_vm_instance] table (unless there are Virtual Machine Instance records without any relationships)

Server [cmdb_ci_server]:

- Cloud:

Server [cmdb_ci_server] -> (Virtualized by::Virtualizes) -> Virtual Machine Instance [cmdb_ci_vm_instance] -> (Hosted on::Hosts) -> Datacenter (one of the [configured cloud datacenters](#)) -> (Hosted on::Hosts) -> Cloud Service Account [cmdb_ci_cloud_service-account]

- Non-Cloud:

Server [cmdb_ci_server] -> Virtual Machine Instance [cmdb_ci_vm_instance] -> Datacenter class (one of the [configured non-cloud datacenters](#))

OR

Server [cmdb_ci_server] has no relationships with Virtual Machine Instance [cmdb_ci_vm_instance]

- Total: Equals the record count in the Server [cmdb_ci_server] table (unless there are Server records without any relationships)

Application [cmdb_ci_appl]:

- Cloud:

Application [cmdb_ci_appl] -> (Runs on::Runs) -> Server [cmdb_ci_server] -> (Virtualized by::Virtualizes) -> Virtual Machine Instance [cmdb_ci_vm_instance] -> (Hosted on::Hosts) -> Datacenter (one of the [configured cloud datacenters](#)) -> (Hosted on::Hosts) -> Cloud Service Account [cmdb_ci_cloud_service-account]

OR

Application [cmdb_ci_appl] -> (Hosted on::Hosts) -> Datacenter class (one of the [configured cloud datacenters](#)) -> (Hosted on::Hosts) -> Cloud Service Account [cmdb_ci_cloud_service-account]

- Non-Cloud:

Application [cmdb_ci_appl] -> Server [cmdb_ci_server] -> Virtual Machine Instance [cmdb_ci_vm_instance] -> Datacenter Datacenter class (one of the [configured non-cloud datacenters](#))

OR

Application [cmdb_ci_appl] -> Server [cmdb_ci_server] -> no relationship with Virtual Machine Instance [cmdb_ci_vm_instance]

- Total: Equals the sum of Cloud + Non-Cloud (not the record count in the Application [cmdb_ci_appl] table)

Kubernetes Cluster [cmdb_ci_kubernetes_cluster]:

- Cloud:

Kubernetes Cluster [cmdb_ci_kubernetes_cluster -> (Hosted on::Hosts) -> Datacenter (cloud logical data center) -> (Hosted on::Hosts) -> Cloud Service Account [cmdb_ci_cloud_service-account]]

- Non-Cloud:

Kubernetes Cluster [cmdb_ci_kubernetes_cluster -> Datacenter (non-cloud logical datacenter)]

OR

Kubernetes Cluster [cmdb_ci_kubernetes_cluster -> no relationship with -> Datacenter]

- Total: Equals the record count in the Kubernetes Cluster [cmdb_ci_kubernetes_cluster] table (unless there are Kubernetes Cluster records without any relationships)

Database Instance [cmdb_ci_db_instance]:

- Cloud:

Database Instance [cmdb_ci_db_instance] -> (Runs on::Runs) -> Server -> (Virtualized by::Virtualizes) -> Virtual Machine Instance [cmdb_ci_vm_instance] -> (Hosted on::Hosts) -> Logical Datacenter (one of the [configured cloud datacenters](#)) -> (Hosted on::Hosts) -> Cloud Service Account [cmdb_ci_cloud_service-account]

OR

Database Instance [cmdb_ci_db_instance] -> Logical Datacenter [cmdb_ci_logical_datacenter] (one of the [configured cloud datacenters](#)) -> (Hosted on::Hosts) -> Cloud Service Account [cmdb_ci_cloud_service-account]

- Non-Cloud:

Database Instance [cmdb_ci_db_instance] -> Server [cmdb_ci_server] -> Virtual Machine Instance [cmdb_ci_vm_instance] -> datacenter (non-cloud logical datacenter)

OR

Database Instance [cmdb_ci_db_instance] -> Server [cmdb_ci_server] with no relationship -> Virtual Machine Instance [cmdb_ci_vm_instance]

- Total: Sum of Cloud + Non-cloud (can be less than total number of records, subtracting the badly created records)

Storage Volume [cmdb_ci_storage_volume]:

- Cloud:

cmdb_ci_storage_volume -> (Hosted on::Hosts) -> Logical Datacenter [cmdb_ci_logical_datacenter] (one of the [configured cloud datacenters](#)) -> (Hosted on::Hosts) -> Cloud Service Account [cmdb_ci_cloud_service-account]

- Non-Cloud:

cmdb_ci_storage_volume -> Logical Datacenter [cmdb_ci_logical_datacenter] (one of the [configured non-cloud datacenters](#))

OR

cmdb_ci_storage_volume -> no relationships with Datacenter

Cloud Object Storage [cmdb_ci_cloud_object_storage]:

- Cloud:

Cloud Object Storage [cmdb_ci_cloud_object_storage] -> (Hosted on::Hosts) -> Logical Datacenter [cmdb_ci_logical_datacenter] -> (Hosted on::Hosts) -> Cloud Service Account [cmdb_ci_cloud_service-account] (requires [CMDB CI Class Models store app](#))

- Non-Cloud:

N/A (This table can never have non cloud records)

Service Accounts [cmdb_ci_cloud_service_account]:

- Cloud:

datacenter_type attribute is populated with correct datacenter class

- Non-Cloud:

N/A (This table can never have non-cloud records)

Edit a related table from CMDB performance insights

Edit a related table on the Related Entries [cmdb_related_entry] table directly from the CMDB performance insights tool in the CMDB Workspace Insights view. Update the related table to correctly reference another CI in the Referenced field when a reference is missing from the related table.

Before you begin

Role required: sn_cmdb_admin (CMDB Admin) and any role needed to access a related table.

About this task

A record is missing a reference when the Referenced field for that record is empty. CMDB performance insights enables you to easily edit related tables that are missing a reference after you drill down on a slice of the Related records missing reference chart.

You can see the full list of Related tables and associated Referenced fields in the Related Entries [cmdb_related_entry] table.

Procedure

1. Navigate to **Workspaces > CMDB Workspace**.
2. In the CMDB Workspace menu bar, select **Insights**.
3. On the CMDB performance insights tile, select **View performance insights**.
4. Navigate to the Related records tile of the Payloads & CIs tab.
5. Select a slice of the chart in the Related records missing reference by table card.
6. Select a check box next to a record.
7. Select **Edit**.
8. Specify applicable CMDB CI references.
9. Select **Update**.

Related concepts

- Configuration Management and the CMDB
- Insights view in CMDB Workspace

Edit a data source from CMDB performance insights

Edit a data source for your Service Graph Connectors to define what data an import set should ingest. Consider configuring batch processing to make ingestion of incoming data more efficient and reduce impacts on the performance of your instance.

Before you begin

Role required: sn_cmdb_admin (CMDB Admin) and import_admin.

Procedure

1. Navigate to **Workspaces > CMDB Workspace**.
2. In the CMDB Workspace menu bar, select **Insights**.
3. On the CMDB performance insights tile, select **View performance insights**.
4. Select the Sources with batch processing turned off card in the Service Graph connectors tab.
5. On the Data sources list view, select a check box next to a record.
6. Select **Edit**.
7. Configure the data source.
For more information about data sources, see:
 - [Create a data source](#)
 - [Data source fields](#)
8. Select **Update**.

Edit a scheduled data import from CMDB performance insights

Edit a scheduled data import directly from CMDB performance insights for your Service Graph Connectors. Consider enabling **Concurrent Import** with a custom size partition to split incoming data into multiple import sets and transform the import sets concurrently to reduce processing time.

Before you begin

Role required: sn_cmdb_admin (CMDB Admin) and one of these roles:

- import_scheduler
- import_admin

Procedure

1. Navigate to **Workspaces > CMDB Workspace**.
2. In the CMDB Workspace menu bar, select **Insights**.
3. On the CMDB performance insights tile, select **View performance insights**.
4. In the Service Graph connectors tab, click the Sources with concurrent import turned off card or the Sources with non-custom size partition method card.
5. In the Scheduled Import sets list view, select a check box next to a record.
6. Select **Edit**.
7. Configure the scheduled data import.
You may need to change your application scope to create or edit a scheduled data import from CMDB Workspace.

For more information about updating a scheduled data import, see [Schedule a data import](#).
8. Select **Update**.

Related concepts

- [Insights view in CMDB Workspace](#)

Components installed with CMDB Workspace

Several types of components are installed with the activation of the CMDB Workspace (sn_cmdb_ws) plugin, including properties, tables, user roles, and scheduled jobs.

Note: The Application Files table lists the components that are installed with this application. For instructions on how to access this table, see [Find components installed with an application](#).

In addition, the CMDB Workspace plugin adds the CMDB Group type 'CMDB Workspace'.

Properties installed

The following properties are installed by the Configuration Management (CMDB) (com.snc.cmdb) plugin which is included in base systems.

Property	Description
sn_cmdb_ws.ci_overview.manage_d_by_me.enabled	<p>Shows/hides the My CIs section of CI Overview on the CMDB Workspace landing page, for users with sn_cmdb_editor role.</p> <ul style="list-style-type: none">• Type: true false• Default: true• Location: Add to System Properties [sys_properties] table.
sn_cmdb_ws.total_cis.enabled	<p>Shows/hides the Total CIs section on the CMDB Workspace landing page.</p> <ul style="list-style-type: none">• Type: true false• Default: true• Location: Add to System Properties [sys_properties] table.
sn_cmdb_ws.ms.discovery_source_not_reporting_max_days	Number of days after which if one or more discovery sources stop reporting CIs, that CI is included in the CIs not reported by discovery

Property	Description
	<p>sources chart in the CMDB 360 view.</p> <ul style="list-style-type: none"> • Type: integer • Default: 7 • Location: Navigate to Workspaces > CMDB Workspace and then select CMDB 360. Select Settings and configure Number of days since CIs were last discovered by a discovery source in the Potential issues section. • Learn more: Configure the CMDB 360 dashboard.
sn_cmdb_ws.ms.report_max_limit	<p>Maximum number of record that appears as list views when drilling down from the following charts:</p> <ul style="list-style-type: none"> • CIs not reported by discovery sources • Data mismatch • CIs by number of discovery sources • CIs with a single source <p>Details:</p> <ul style="list-style-type: none"> • Type: integer • Default: 100,000 • Location: Navigate to Workspaces > CMDB Workspace and then select CMDB 360.

Property	Description
	<p>Select Settings and configure Maximum number of records to process in the Global section.</p> <ul style="list-style-type: none"> Learn more: Configure the CMDB 360 dashboard.
sn_cmdb_ws.unifiedmap.map_search_filter.default_levels	<p>Initial default number of levels from the home node, up and down the CMDB hierarchy, to show on a map. Up the hierarchy goes from all parents to their parents up to the specified level. Down the hierarchy goes from all children to their children up to the specified level.</p> <ul style="list-style-type: none"> Type: integer Default: 3 Location: Add to System Properties [sys_properties] table. <p>Editing affects all users and requires the sn_cmdb_admin user role.</p>
sn_cmdb_ws.unifiedmap.map_search_filter.max_levels	<p>Maximum number of levels from the home node, up and down the CMDB hierarchy, to show on a map. Up the hierarchy goes from all parents to their parents up to the specified level. Down the hierarchy goes from all children to their children up to the specified level.</p> <ul style="list-style-type: none"> Type: integer

Property	Description
	<ul style="list-style-type: none"> • Default: 25 • Location: Add to System Properties [sys_properties] table. <p>Editing affects all users and requires the sn_cmdb_admin user role.</p>

Roles installed

The following roles are installed by the Configuration Management (CMDB) (com.snc.cmdb) plugin which is included in base systems. These roles are required for access and interaction with the CMDB Workspace, and are included for completeness.

Role title [name]	Description	Contains roles
CMDB Admin [sn_cmdb_admin]	Provides full access to CMDB data, tools, and UIs within CMDB Workspace. A CMDB Admin, for example, sets policies in the CI Class Manager and application service requirements. CMDB Admin provides the highest level of access to the CMDB.	<ul style="list-style-type: none"> • data_manager_admin • canvas_admin • sn_cmdb_editor • cmdb_ms_admin
CMDB Editor [sn_cmdb_editor]	Provides access to CMDB records within CMDB Workspace. A CMDB Editor has writing privileges to CMDB Data Manager tasks and to CIs but can't change policies	<ul style="list-style-type: none"> • sn_cmdb_user • cmdb_ms_editor

Role title [name]	Description	Contains roles
	such as in the CMDB Data Manager or in the CI Class Manager.	
CMDB User [sn_cmdb_user]	Provides read-only access to the CMDB data and to basic UIs such as CMDB reports and dashboards within CMDB Workspace.	<ul style="list-style-type: none"> • canvas_user • data_manager_user • cmdb_ms_user
Multisource CMDB Admin [cmdb_ms_admin]	Provides full access to data, dashboard configurations, and queries related to Multisource CMDB and CMDB 360.	cmdb_ms_editor
Multisource CMDB Editor [cmdb_ms_editor]	Provides access to CMDB 360 records and enables users to create and edit CMDB 360 queries.	cmdb_ms_user
Multisource CMDB User [cmdb_ms_user]	Provides read-only access to multisource data, CMDB 360 queries, and to related UIs such as CMDB 360 reports and dashboards, and Multisource Report Builder.	cmdb_read

Scheduled jobs installed

Scheduled job	Description
CMDB Workspace Collection	Updates the information in all the 7-Day Activity charts such as the CI Activity in Last 7 Days chart.
Multisource Dashboard Analytics Population	Runs daily to calculate the aggregate statistics of CMDB 360 data and populates the CMDB 360 dashboard landing page.
Multisource Dashboard Collection	Collects CMDB 360 data and then invokes the Multisource Dashboard Analytics Population scheduled job.
Multisource Dashboard Data Generator (Demo data)	Demo job that populates CMDB 360 tables with simulated data, to showcase the CMDB 360 dashboard landing page. Runs on demand.
insight – cloud non-cloud aggregate	Runs every 30 minutes to collect data for the Cloud and non-cloud resources chart.
Insight - Cloud Applications	Collects cloud data for the cmdb_ci_appl table.
Insight - Non-cloud Applications	Collects non-cloud data for the cmdb_ci_appl table.
Insight - Cloud Object Storage	Collects cloud data for the cmdb_ci_cloud_object_storage table.
Insight - Cloud Service Account	Collects cloud data for the cmdb_ci_cloud_service_account table.

Scheduled job	Description
Insight - Cloud DB Instances	Collects cloud data for the cmdb_ci_db_instance table.
Insight - Non-cloud DB Instances	Collects non-cloud data for the cmdb_ci_db_instance table.
Insight - Cloud Kubernetes Cluster	Collects cloud data for the cmdb_ci_kubernetes_cluster table.
Insight - Non-cloud Kubernetes Cluster	Collects non-cloud data for the cmdb_ci_kubernetes_cluster table.
Insight - Cloud Servers	Collects cloud data for the cmdb_ci_server table.
Insight - Non-cloud Servers	Collects non-cloud data for the cmdb_ci_server table.
Insight - Cloud Storage Volume	Collects cloud data for the cmdb_ci_storage_volume table.
Insight - Non-cloud Storage Volume	Collects non-cloud data for the cmdb_ci_storage_volume table.
Insight - Cloud VM Instances	Collects cloud data for the cmdb_ci_vm_instance table.
Insight - Non-cloud VM Instances	Collects non-cloud data for the cmdb_ci_vm_instance table.
CMDB Workspace – Populate aggregates Daily (Renamed from CMDB Workspace - Group and Encoded Query Counts)	<p>Runs daily to collect the latest data from the instance for cards such as the CI Summary chart in the CMDB Workspace landing page, and the following cards in the Insights view:</p> <ul style="list-style-type: none"> • CIs processed by IRE • CIs used in Data Manager policies

Scheduled job	Description
	<ul style="list-style-type: none"> • CIs used in data attestation • Query Builder queries • Intelligent search <p>Populates the Base Aggregate Data [sn_cmdb_ws_base_aggregate_data] table with the collected data.</p> <p>Once collection completes, invokes the CMDB Workspace Aggregates Daily Collection scheduled job.</p>
CMDB Workspace – Populate aggregates Monthly	Runs monthly to collect data from the instance for the 'CIs processed by IRE based on source' card in the Insights view. Once collection completes, invokes the CMDB Workspace Aggregates Monthly Collection scheduled job.
CMDB Workspace Aggregates Daily Collection	A Performance Analytics job that stores the latest data generated by the CMDB Workspace – Populate aggregates Daily scheduled job. The stored data is then shown in respective cards in the Insights view.
CMDB Workspace Aggregates Monthly Collection	A Performance Analytics job that stores the latest data generated by the CMDB Workspace – Populate aggregates Monthly scheduled job. The stored data is then shown in respective cards in the Insights view.

Scheduled job	Description
[Demo] — CMDB Workspace Collection	<p>Supports demo data for CMDB Workspace charts in various views such as the Home view.</p> <p>Installed only if Load demo data was checked when the CMDB Workspace store app was installed or upgraded.</p>
[Demo] — CMDB Workspace demo data	<p>Activates demo data for CMDB Workspace charts in views such as the Insights view, and CMDB 360 view (if CMDB 360 is enabled).</p> <p>Populates the Base Aggregate Data [sn_cmdb_ws_base_aggregate_data] table with random numbers that illustrate trend lines in charts in the Home and Insights views. It then runs demo data collection scheduled jobs.</p> <p>Installed only if Load demo data was checked when the CMDB Workspace store app was installed or upgraded.</p>
[Demo] CMDB Workspace Aggregates Daily Collection	<p>A Performance Analytics job that supports demo data for CMDB Workspace charts in various views such as the Insights view and which is installed only if Load demo data was checked when the CMDB Workspace store app was installed or upgraded.</p>

Scheduled job	Description
	<p>Stores the demo data generated by the CMDB Workspace – Populate aggregates Daily scheduled job. The demo data is then shown in respective cards in the Insights view.</p>
<p>[Demo] CMDB Workspace Aggregates Monthly Collection</p>	<p>A Performance Analytics job that supports demo data for CMDB Workspace charts in various views such as the Insights view and which is installed only if Load demo data was checked when the CMDB Workspace store app was installed or upgraded.</p> <p>Stores the demo data generated by the CMDB Workspace – Populate aggregates Monthly scheduled job. The demo data is then shown in respective cards in the Insights view.</p>
<p>CMDB Workspace Collection On Demand With Lookback</p>	<p>Collects historical data (past 30 days) for some indicators such as the Integration outliers.</p> <p>Runs automatically during install or upgrade and isn't configured with any recurring schedule.</p>

Tables installed

Table	Description
CMDB Group Metadata [sn_cmdb_ws_group_metadata]	CMDB group data with count of Cls in each group.
CMDB Group Query Metadata [sn_cmdb_ws_group_query_metadata]	Encoded queries with count of Cls in each encoded query subgroup.
NLQ Sample Search [sn_cmdb_ws_nlq_sample_search]	Sample searches that appear when selecting the search box of an Intelligent Search for CMDB widget.
NLQ Sample Search Table [sn_cmdb_ws_nlq_sample_search_table]	References the actual tables associated with sample searches that appear when selecting the search box of an Intelligent Search for CMDB widget.
NLQ Excluded Table [sn_cmdb_ws_nlq_excluded_table]	Tables with ambiguous names that should be excluded from Intelligent Search. These tables won't appear in suggestions, and other tables with similar names are given higher priority in search results.
Important Actions Configuration	Configuration of appearance and behavior of Important action cards that appear on the landing page of the CMDB Workspace. Editing in this table is

Table	Description
[sn_cmdb_ws_imp_action_card_config]	accessible only to users with the sn_cmdb_admin role. Authorized users can modify attributes of an important action such as Active and Filter conditions, but can't modify the Type, Persona, and Table attributes.
Quick Links [sn_cmdb_ws_quick_links]	Quick links that appear on the CMDB Workspace landing page.
CIs by Number of Sources [sn_cmdb_ws_ms_cis_by_number_of_sources]	CIs grouped by the number of discovery sources they were discovered by. Records appear as a list view when clicking on the CIs by number of discovery sources bar chart in the CMDB 360 view.
CIs With a Single Source [sn_cmdb_ws_ms_cis_with_single_source]	CIs reported by only one discovery source. Records appear as a list view when selecting the CIs with a single discovery source chart in the CMDB 360 view.
CMDB Workspace Multisource Class Metadata [sn_cmdb_ws_ms_class_metadata]	Class configuration settings such as class selections and weights, that CMDB Admins can modify from the CMDB 360 view. Used by CMDB 360 analytics population queries to store a sample set of coverage data for the CIs by number of sources and CIs with a single source drill-downs cards.
Multisource Data Mismatch Records	Data mismatch records found from attribute discrepancies from various data sources. Records appear as a list view when

Table	Description
[sn_cmdb_ws_ms_data_mismatch]	selecting the Data mismatch chart in the CMDB 360 view.
CMDB Workspace Multisource Data Mismatch Configurations [sn_cmdb_ws_ms_data_mismatch_config]	Data mismatch settings, such as the selection of classes, attributes, conditions, and weights, that CMDB Admins can modify in the CMDB 360 view. Used by CMDB 360 analytics population queries to store a sample set of data mismatch records in the sn_cmdb_ws_ms_data_mismatch table.
Discovery sources not reporting CIs [sn_cmdb_ws_ms_discovery_sources_not_reporting]	Data for when one or more discovery sources stop reporting CIs after the specific X number of days (X can be set in the CMDB 360 view and is stored in the sn_cmdb_ws.ms.discovery_source_not_reporting_max_days system property). Records appear as a list view when selecting the CIs not reported by discovery sources chart in the CMDB 360 view.
Multisource Workspace Settings Weight Type [sn_cmdb_ws_ms_settings_weight_type]	Weight type (automatic or manual) for data mismatch and coverage settings in CMDB 360 view.
Datacenter Types [sn_cmdb_ws_datacenter_type]	Classifications of datacenter classes which are considered cloud or non-cloud in the organization, which is used for the Cloud and non-cloud resources chart.

Table	Description
CMDB Insight Query Categories [sn_cmdb_ws_insight_query_category]	Class categories and their status for inclusion in the Cloud and non-cloud resources chart.
CMDB Insight Data [sn_cmdb_ws_insight_data]	Legacy table for storing data for the Cloud and non-cloud resources chart. Starting with CMDB Workspace version 3.4, the CMDB Product Insight Data [sn_cmdb_ws_product_insight_data] table is used instead.
Base Aggregate Data [sn_cmdb_ws_base_aggregate_data]	Aggregation data for CMDB Workspace cards and parent table to other aggregation tables. When processing for a count completes, State of the existing record for that count is set to retired and a new record for that count is created with the updated count value. Cards in CMDB Workspace views show values only for counts where State = ready . Read only.
Service Graph Connector [sn_cmdb_ws_service_graph_connector]	Details about all Service Graph Connectors that are currently available in the ServiceNow Store (excluding any Innovation Lab connectors). Data is used to provide a current count for the

Table	Description
	<p>Service Graph Connectors card in the Insights view.</p> <p>Read only.</p>
Rating Configuration <code>[sn_cmdb_ws_rating_config]</code>	<p>Start and end values for different ratings of a category. For example, start and end values for low, moderate, and high adoption level ratings for the data governance category in the Insights view.</p> <p>Read only.</p>
Feature Category <code>[sn_cmdb_ws_feature_category]</code>	<p>Metadata of the feature card for most of the features at CMDB Workspace level.</p> <p>Read only.</p>
CMDB Product Insight Data <code>[sn_cmdb_ws_product_insight_data]</code>	<p>Aggregated count for each class and datacenter class, used for cards in the Insights view.</p> <p>Read only.</p>
Feature Category Runtime Attributes <code>[sn_cmdb_ws_feature_category_runtime_attribute]</code>	<p>Runtime attributes related to performance insights, that can be used when running various scheduled jobs.</p> <p>Read only.</p>

Table	Description
Integration Aggregate Data [sn_cmdb_ws_integration_aggregate_data]	<p>Integration outlier information related to performance insights. Details for each Integration with execution records which contain processing rate or rows outliers.</p> <p>Read only.</p>
Partial Payload Items [sn_cmdb_ws_partial_payload_item]	<p>Extracted partial payload information related to performance insights. Includes discovery source, error, and class, associated with each partial payload item.</p> <p>Read only.</p>
Application Service Insights Data [sn_cmdb_ws_app_svc_insight_data]	<p>Application services details such as classification (cloud vs. non-cloud, hybrid, or unknown) and count of CIs associated with the application service.</p> <p>Read only.</p>
Table Attributes [sn_cmdb_ws_node_map_table_attributes]	<p>Set of extended properties, per class, that appear in the Attributes pane in the contextual side panel for a selected CI in Unified Map.</p> <p>For more information, see Configure extended properties for a class.</p>

Table	Description
Node Map Related Items [sn_cmdb_ws_node_map_related_item]	<p>Categorization and order of related items, per class, that appear in the Related items pane in the contextual side panel, in CI badges, and in timelines. Applies in Unified Map.</p> <p>For more information, see Configure related items.</p>
Node Map References [sn_cmdb_ws_node_map_references]	<p>Reference relationships that appear as a dotted line relationship in Unified Map.</p> <p>For more information, see Configure map references.</p>
Class Icons [sn_cmdb_ws_class_icon]	Maps ServiceNow platform UI Builder (UIB) icon values (for example 'serverbox-outline') to CMDB class display names that they are used for (for example 'Network Disk') in Unified Map.
Node Map Profiles [sn_cmdb_ws_node_map_profiles]	<p>Class profiles that include default filters and some Unified Map settings, per class. A class profile is applied if no filter preset is in effect, to the initial map and when setting the filter preset to Default view.</p> <p>Specifying a class profile can be especially useful for Service</p>

Table	Description
	Mapping data with the Mapped Application Service class. For more information, see Configure class profiles .

CMDB Identification and Reconciliation

The Identification and Reconciliation module provides a centralized framework for identifying and reconciling data from different data sources. It helps maintain the integrity of the CMDB and some non-CMDB tables when multiple data sources are used to create and update CI records.

The use of multiple data sources increases the risk of introducing inconsistencies through duplicate records. To maintain the integrity of the database, it is important to correctly identify CIs and services so that new records are created only for CIs that are truly new.

The Identification and Reconciliation Engine (IRE) helps maintain the data integrity as follows:

- Prevent duplicate CIs by uniquely identifying CIs using sets of identification rules
- Reconcile CI attributes by allowing only authoritative data sources to write to the CMDB or to a supported non-CMDB table
- Reclassify CIs
- Provide a centralized framework to perform identification and reconciliation across different data sources

Data sources such as ServiceNow Event Management, Horizontal Discovery, Import Sets, Cloud Insights, Pattern Discovery, and Manual Entry, use IRE APIs to perform CI identification and reconciliation. In addition, any 3rd party data source can leverage REST/Scriptable IRE APIs to also perform CI identification and reconciliation.

Support for non-CMDB tables

IRE processes support some non-CMDB tables. You can create identification rules, reconciliation rules, and other IRE-related rules to ensure the integrity of data inserted or updated in supported non-CMDB tables. For details, see [IRE support for non-CMDB tables](#).

- [Domain separation and CMDB Identification and Reconciliation](#)

Domain separation is supported in the CMDB Identification and Reconciliation feature. Domain separation enables you to separate data, processes, and administrative tasks into logical groupings called domains. You can control several aspects of this separation, including which users can see and access data.

- [Components and process of Identification and Reconciliation](#)

The CMDB Identification and Reconciliation functionality is supported by the Identification and Reconciliation engine (IRE), rules, and tasks. Identification rules, reconciliation rules, IRE data source rules, deduplication tasks, and reclassification tasks determine how IRE identifies and reconciles CI.

- [Apply CI Identification and Reconciliation to Import Sets](#)

You can apply CMDB Identification and Reconciliation processes when Import Sets are used to import CIs into the CMDB. CI identification can prevent duplicate CIs in the CMDB, which Import Sets might otherwise cause.

- [Identification rules](#)

The CMDB identification process relies on identification rules to uniquely identify CIs.

- [Reconciliation rules](#)

Reconciliation rules determine which discovery sources can update CI attributes.

- [Create an IRE data source rule](#)

When using Identification and Reconciliation Engine (IRE), you can prevent a specific discovery (data) source from inserting new CIs for

a specific class. Create IRE data source rules for discovery sources that you don't trust in creating CIs but continue to trust in updating those CIs that exist.

- **Detecting duplicate CIs**

When the identification process encounters duplicate CIs, it groups each set of duplicate CIs into a de-duplication task for review and remediation. A large number of duplicate CIs might be due to weak identification rules. You can configure the identification engine to reconcile duplicate CIs.

- **Generate and simulate payload execution using identification simulation**

Identification simulation is a central location for automatically constructing a payload that is guaranteed to be complete and valid. You can then simulate the processing of the payload by the identification and reconciliation engine (IRE) and examine the results before actually submitting it for execution by IRE.

- **CI reclassification during IRE processing**

During the Identification and Reconciliation Engine (IRE) CI identification process, a CI might need to be reclassified to a different sys_class_name type. By default, CIs are reclassified automatically. If automatic reclassification is disabled, then the CI is not reclassified and the system generates a reclassification task for your review.

- **CMDB dependent relationship rules**

Service definitions consist of CI types and relationship types. Dependent relationship rules define the dependency structure of the CI types and the relationship types in these service definitions, helping in CI identification and in the construction of business service maps.

- **IRE support for non-CMDB tables**

Apply Identification and Reconciliation Engine (IRE) processes to supported non-CMDB tables to ensure data integrity and health of those tables.

- **Effective usage of CMDB Identification**

Use CMDB Identification effectively.

- [Properties for Identification and Reconciliation](#)

Use the Identification and Reconciliation properties to configure the identification and reconciliation engine (IRE).

- [Components installed with Identification and Reconciliation](#)

Several types of components are installed with Identification and Reconciliation (included in the com.snc.cmdb plugin), including tables.

Related concepts

- [Relation qualifier](#)

Domain separation and CMDB Identification and Reconciliation

Domain separation is supported in the CMDB Identification and Reconciliation feature. Domain separation enables you to separate data, processes, and administrative tasks into logical groupings called domains. You can control several aspects of this separation, including which users can see and access data.

Overview

Domain separation is enforced during the [CMDB Identification and Reconciliation](#) (IRE) process. IRE processes are domain aware and domain separation is applied to the Identification and Reconciliation rules.

For more information about domain separation, see [domain separation](#).

How domain separation works in Identification and Reconciliation

Domain separation in the identification engine is enforced when users activate the domain separation plugin. Domain separation for IRE has two modes of operation in domain separated instances:

- Strict mode (enabled by default): In this mode, identification processes only those CIs in which the domain ID is identical to the domain of the currently logged in user. If duplicate CIs exist across domains (including

parent and child domains), then those CIs aren't considered duplicate CIs because their domain IDs don't match.

-

Platform domain separation mode (disabled by default): In this mode, IRE follows the platform domain separation behavior. So during identification, parent domains can access all CIs within their child domains or any of the domains it has visibility into. For more information, see [Visibility domains and Contains domains](#).

Platform domain separation mode is intended to be used by advanced users for very specific or advanced use cases.

Note:

Platform domain separation mode introduces some risks that are greater on upgraded instances and much lesser on zBooted instances.

Depending on how IRE processes are configured on a domain separated instance, setting IRE to use platform domain separation mode might result in unexpected and undesirable behavior if not used carefully. One of the risks is if enabling platform domain separation mode is followed by running IRE processes from a different domain than the one on which IRE processes were previously run. In this situation, CIs that were previously identified as unique, might get identified as duplicate CIs and might cause some applications to start failing.

If any application is already using IRE effectively in domain separated environment, then there's no advantage in switching to platform domain separation mode that might create some risk.

Use the

[glide.identification_engine.platform_domain_separation_enabled](#) system property to switch between those two modes for IRE domain separation. By default, this property is set to **false**.

Platform domain separation mode

Set the system property

`glide.identification_engine.platform_domain_separation_enabled` to `true` to enable the platform domain separation mode for IRE processing. With the platform domain separation mode, parent domains can access all of their child domains during IRE processing. For example, IRE can detect a matching CI in a child domain and then update that CI instead of creating a new one.

In the platform domain separation mode for IRE:

- IRE run from a parent domain can access CIs contained within their domain, child domains that are lower in the domain hierarchy, and global domain.
- IRE run from the global domain can access all CIs.
- [Visibility domains](#) and [Contains domains](#) are supported.

Note: When platform domain separation mode is enabled, there might be a sudden increase in IRE detection of duplicate CIs.

Domain separation during the Identification Process

Domain separation during the Identification process is enforced as follows:

- Regardless of the setting of the `glide.identification_engine.platform_domain_separation_enabled` system property:
 - Domain IDs don't need to be explicitly sent in the input payload of the identification engine APIs. Internally, the identification engine causes the current domain ID of the user to call the identification engine APIs.
 - During matching, if no records are found and a CI is inserted, the CI domain ID is the same as the domain ID of the logged-in user's domain. When updating a CI, the CI domain ID doesn't change.
 - During matching, if duplicates are found, De-Duplication tasks created in the `[reconcile_duplicate_task]` table have the same domain ID as of the duplicate CIs.

- During matching, if reclassification of the CI isn't allowed, reclassification tasks are created in the [reclassification_task] table, with the same domain ID as the CI for which reclassification is needed.
- When the system property `glide.identification_engine.platform_domain_separation_enabled` is set to **false**:
 - Only CIs that have the same domain ID as the currently logged-in user's domain are used during matching.
 - Duplicate CIs that exist across domains (including parent and child domains) aren't considered as duplicate CIs by IRE.
- When the system property `glide.identification_engine.platform_domain_separation_enabled` is set to **true**:
 - Duplicate CIs that exist across domains (such as parent and child domains) are considered as duplicate CIs by IRE.
 - CIs from the logged in user domain and child domains are used during matching.

Domain separation and Identification Rules

The identification rules and identification inclusion rules used during the identification process are always defined at the global level. For example, the following tables don't have a `sys_domain` field:

- Identifier (cmdb_identifier)
- Identifier Entries (cmdb_identifier_entry)
- Related Entries (cmdb_related_entry)
- Identification Inclusion Rules (cmdb_ie_active_config)

Domain separation and Reconciliation Rules

The reconciliation definition rules that are used during the reconciliation process can be defined for different domains. For example, the following tables do have `sys_domain`, `sys_overrides`, `sys_domain_path` fields:

- Reconciliation Definition (cmdb_reconciliation_definition)

- Datasource Precedence (cmdb_datasource_precedence)
- Data Source Staleness Definitions (cmdb_datasource_staleness)

Components and process of Identification and Reconciliation

The CMDB Identification and Reconciliation functionality is supported by the Identification and Reconciliation engine (IRE), rules, and tasks. Identification rules, reconciliation rules, IRE data source rules, deduplication tasks, and reclassification tasks determine how IRE identifies and reconciles CI.

Concepts and Components of Identification and Reconciliation

Identification

Process of uniquely identifying CIs, to determine if a CI already exists in the CMDB or if it is a newly discovered CI that must be added to the CMDB. Identification processes rely on [identification rules](#), or on unique IDs for CIs that data sources can provide.

Reconciliation

Process of reconciling CIs and CI attributes by allowing only designated authoritative data sources to write to the CMDB at the CI table and attribute level. The CMDB is updated in real time as records are being processed. There is no staging area to verify the reconciliation activities before they are committed. Reconciliation processes rely on [reconciliation rules](#) and [IRE data source rules](#).

Reconciliation is required only for update operations, when the identification process identifies a CI in the CMDB that matches an incoming CI in the payload. When IRE inserts a new CI, reconciliation is not applied.

De-duplication tasks

If the instance encounters duplicate CIs during the Identification and Reconciliation process, it groups each set of duplicate CIs into a [de-duplication task](#). Review the information in these tasks to see how it was determined that these CIs are duplicates.

Reclassification tasks

During the CI identification process, a matched CI might need to be upgraded, downgraded, or switched to another CI class. If automatic reclassification is disabled, then the system generates a [reclassification task](#). Review the information in these tasks, and decide whether a manual reclassification of the CI is appropriate.

APIs

The Identification and Reconciliation APIs are a centralized set of APIs that can be used with different sources of data such as Discovery, Monitoring, or Import Sets. You can use it to enforce Identification and Reconciliation before data is stored in the CMDB. Data sources do not directly write to the CMDB. Instead, they call APIs first to ensure that the data being written does not introduce inconsistencies.

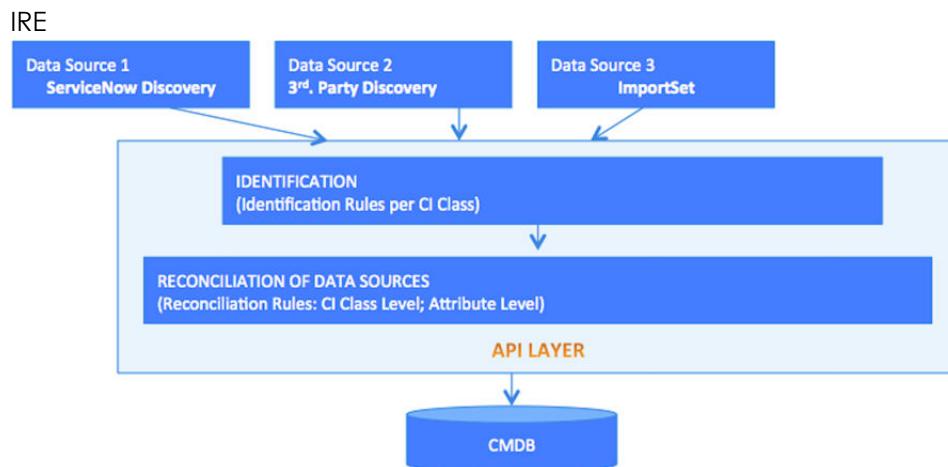
Identification engine APIs are accessible in scoped apps. The Configuration Management For Scoped Apps (CMDB) plugin (com.snc.cmdb.scoped) allows a scoped app in scripts to use the prefix 'sn_cmdb.IdentificationEngine.<method>' to access identification engine APIs. The Configuration Management For Scoped Apps (CMDB) plugin is activated in base systems.

- [createOrUpdateCI\(\)](#): A scriptable API that creates or updates a CI based on identification and reconciliation rules.
- [identifyCI\(\)](#): Similar to the createOrUpdateCI API, but does not commit the result to the database. Use this API with a given payload to find out if the identification engine will perform insert or update operations, without committing the operation.
- [CreateOrUpdateCIEnhanced\(\)](#): A scriptable API that provides the functionality of enhanced IRE features such as partial payload, partial commit, incomplete payload, and deduplication of payload items. You can select the enhanced features to use. However, if you enable partial payloads, then deduplication of payload items and partial commit are automatically enabled.
- [identifyCIEnhanced](#): Similar to the createOrUpdateCIEnhanced API, but does not commit the result to the database. Use this API with a given payload to find out if the identification engine will perform insert or update operations, without committing the operation.

- **CMDBTransformUtil:** An API to be used exclusively with Import Sets to apply Identification and Reconciliation processes to data imported by Import Sets.

Predefined identification are included for many of the tables in the base system. You can customize these rules for your organization. When a new table is created in the CMDB, it derives identification and reconciliation rules from its parent table if these rules exist. To apply identification and reconciliation rules to a new table, create the rules either at the child level or at its parent level.

Process flow of Identification and Reconciliation



Identification and Reconciliation Engine (IRE)

Identification and Reconciliation engine (IRE) is a rule-based engine, operating as an underlying key component in Identification and Reconciliation. IRE provides a centralized framework to perform identification and reconciliation processes across different data sources. IRE uses identification rules, reconciliation rules, and IRE data source rules when processing incoming data before inserting or updating data in the CMDB. IRE processes help maintain data integrity in the CMDB.

- IRE prevents duplicate CIs by uniquely identifying CIs.

- IRE reconciles CI attributes by allowing only authoritative data sources to write to CMDB.

IRE is an underlying key component in Identification and Reconciliation, providing a centralized framework to perform identification and reconciliation processes across different data sources. IRE uses identification rules, reconciliation rules, and IRE data source rules when processing incoming data before inserting that data to the CMDB.

IRE processes help maintain data integrity in the CMDB.

- IRE prevents duplicate CIs by uniquely identifying CIs.
- IRE reconciles CI attributes by allowing only authoritative data sources to write to CMDB.

ServiceNow® applications such as Service Mapping, horizontal discovery, and pattern discovery, use APIs to apply identification and reconciliation processes. You can also apply IRE processes to data imported by import sets. When using other data sources including third-party data sources, you can leverage REST or scriptable IRE APIs to perform identification and reconciliation.

Additional information:

- About properties that affect some functions of IRE: See [Properties for Identification and Reconciliation](#).
- About using IRE APIs: See [IdentificationEngine - Scoped](#).
- About enabling debugging and checking on issues with a payload: See [How to log the payload sent to IRE and check the issues with the payload \[KB0750382\]](#).
- About the steps that IRE performs illustrating how IRE works, such as validating the payload, applying reconciliation, and committing the data to the CMDB: See [\[CMDB - IRE\] How the CMDB Identification and Reconciliation Engine works when passing a CI \(as payload\) to the createOrUpdateCI\(\) \[KB0750386\]](#).
- About how to run a payload through IRE: See [\[CMDB IRE\] How to run the CI identification on demand using the payload \[KB0750383\]](#).

CI identification

The CMDB identification process relies on identification rules to uniquely identify CIs. When possible, CIs can also be uniquely identified using source_name and source_native_key values provided in the sys_object_source_info section of the payload, and the Source [sys_object_source] table. If identification is successful using that method, then it is not necessary to apply matching algorithms that rely on identification rules, which is a slower identification method.

A unique CI identifier can be provided in the optional sys_object_source_info object in the IRE payload.

```
{  
  "items": [  
    {  
      "className": "cmdb_ci_win_server",  
      "values": {  
        "name": "SAMPLABVM52"  
      },  
      "sys_object_source_info": {  
        "source_native_key": "16777219",  
        "source_name": "SCCM",  
        "source_feed": "SCCM Computer Identity",  
        "source_recency_timestamp": "2019-08-26 13:00  
:00"  
      }  
    }  
  ]  
}
```

Payload items identification

IRE generates identifier keys for all payload items in an incoming payload and then uses those keys when trying to match partial and incoming payloads. An identifier key is based either on:

- Combination of the source_name and source_native_key values from the sys_object_source_info object.
- Identification criterion attributes.

IRE stores the identifier keys associated with partial items in the CMDB IRE Partial Payloads Index [cmdb_ire_partial_payloads_index] table, and

then uses those keys to try to match with identifier keys of incoming payloads.

Timestamps in key attributes

To help resolve conflicting attribute values, IRE uses timestamps in the following attributes to identify records that are older than the current record and therefore can be ignored:

-

Most recent discovery (`last_discovered`) and Discovery source (`discovery_source`):

Most recent discovery (`last_discovered`) is the timestamp of when the CI was last discovered. IRE always updates CIs' `last_discovered` and `discovery_source` attributes during payload processing, even when no other CI attributes are updated. When `last_discovered` is provided in the payload, IRE updates the CI with the provided value only if the `last_discovered` time in the payload is newer than the one in the CMDB. If `last_discovered` is not provided in the payload, IRE updates the `last_discovered` attribute with the current timestamp.

You can use the `glide.identification_engine.skip_updating_source_last_discovered_if_older` and the `glide.identification_engine.ire_message_listener_skip_updating_source_last_discovered_to_now` system properties to modify this default behaviour.

-

First discovered (`first_discovered`) is the timestamp of when the CI was first created.

- When the CI is first created: If a value is provided in the payload, IRE inserts that value. Otherwise, IRE inserts the current timestamp.
- In subsequent updates: If a value is provided, IRE updates the CI with the provided value. Otherwise, the attribute is not updated.

You can also use the following system properties to modify how IRE uses the `source_recency_timestamp` value in a payload to update the `last_scan` attribute in the Source [`sys_object_source`] table:

- `glide.identification_engine.skip_updating_last_scan_if_older`

- `glide.identification_engine.ire_message_listener_skip_updating_last_scan_to_now`

Enhanced IRE features

The `CreateOrUpdateCIEnhanced()` and `identifyCIEnhanced` scriptable APIs provides the functionality for the following enhanced IRE features, which can be enabled or disabled as needed:

Partial payloads

IRE isolates items for which data sources did not provide enough information to uniquely identify the CI and therefore processing cannot continue. Some of these items are identified as partial items, which get stored for potential later processing. Other items are identified as incomplete items, which get stored for logging purposes only.

For example: SCCM has multiple feeds such as a disk feed and a computer feed. The disk feed might have complete information about the disk but insufficient information about the computer CI that it depends on.

API option: `partial_payloads` which is enabled by default. When `partial_payloads` is enabled, `partial_commits` and `deduplicate_payloads` are automatically enabled regardless of their setting in options.

Partial commits

Errors in some items do not prevent committing the rest of the items in a payload. Therefore, when a payload contains items with errors, IRE still commits the remaining valid items in the payload. In this situation, some of the uncommitted items are saved as partial payloads and other uncommitted items are saved as incomplete payloads.

API option: `partial_commits` which is enabled by default.

Deduplicate payload items

IRE deduplicates duplicate items within the payload, merging those duplicates into a single payload item for processing.

API option: `deduplicate_payloads` which is enabled by default.

Generate summary

IRE generates summaries in the output payload with processing details such as the number of updates per class.

API option: generate_summary which is disabled by default.

Partial items

A payload item is determined to be a partial item if it contains the necessary data for unique identification and if it has one of the following errors. Unique identification requires that the payload item has the sys_object_source_info section with source_name and source_native_key values, or the full set of the identification criterion attributes specified for the CI class, or both.

IRE errors for a partial item:

- MISSING_MATCHING_ATTRIBUTES — Item does not have identification criterion attributes to use at least one identifier entry for matching.
- REQUIRED_ATTRIBUTE_EMPTY — Unable to create a CI because a required attribute is missing.
- MISSING_DEPENDENCY — Dependent CI is missing a dependency relation which is specified in the payload.
- INSERT_NOT_ALLOWED_FOR_SOURCE — An IRE data source rule prevents the specified data sources from creating CIs of the specified class.

For more details about IRE error messages, see [IRE error messages](#).

If processing fails because payload items are determined to be partial items, then the partial items are saved as partial payloads in the CMDB IRE Partial Payloads [cmdb_ire_partial_payloads] table in JSON format for later potential processing. IRE uses identifier keys to attempt to match incoming payloads with stored partial payloads.

If later, a data source sends the data that was missing in the partial item, IRE matches the incoming payload with the stored partial payloads. IRE then merges any matching partial payloads with the incoming payload. To resolve any conflicting attributes, IRE uses either source_recency_timestamp (when source_native_key and source_name are identical), or static reconciliation rules specified for the class. The

result is a complete and valid payload which IRE then processes to create or update the respective CIs.

Partial payloads older than 90 days are deleted from the CMDB IRE Partial Payloads [cmdb_ire_partial_payloads] table.

Sample of a partial payload:

```
Disk feed:  
{  
    "items": [  
        {  
            "className": "cmdb_ci_computer",  
            "sys_object_source_info": {  
                "source_native_key": "Server001",  
                "source_name": "SCCM",  
                "source_feed": "DISK_FEED",  
                "source_recency_timestamp": "2019-08-26 13:00  
:00"  
            }  
        },  
        {  
            "className": "cmdb_ci_disk",  
            "values": {  
                "name": "disk1"  
            }  
        }  
    ],  
    "relations": [{  
        "parent": 0,  
        "child": 1,  
        "type": "Contains::Contained by"  
    }]  
}
```

The computer item in the above payload does not have any attributes and therefore IRE can't process it. However, source_name and source_native_key are provided making it a partial item. Because the computer item is partial, the disk item that depends on the computer item, is a partial item too.

Sample of a subsequent payload that completes the previous partial payload by providing the missing details:

```
Server/Computer feed:  
{
```

```
"items": [
  {
    "className": "cmdb_ci_linux_server",
    "values": { "name": 'linux001',
                "ip_address": "100.126.38.19",
                "mac_address": "DSWER4587" },
    "sys_object_source_info": {
      "source_native_key": "Server001",
      "source_name": "SCCM",
      "source_feed": "COMPUTER_IDENTITY",
      "source_recency_timestamp": "2019-08-26 14:00
:00"
    }
  }
]
```

The computer in the partial payload and the server in the new payload match because they have identical source_name and source_native_key. Therefore, the partial payload and the new payload are merged, the operation is committed, and the partial payload is deleted from the Partial Payloads table.

There is a limit on the number of items per partial payload, which is set by the [glide.identification_engine.partial_payload_items_max_size](#) property (1000 by default). Storing associated relationships, references, and dependent items, in one partial payload, can result in reaching that limit, in which case, the payload is split into multiple partial payloads.

For more information about partial payloads, see [CreateOrUpdateCIEnhanced\(\)](#).

Incomplete items

A payload item is determined to be an incomplete item if:

- It does not contain all the data necessary for unique identification
- It has an error that is not associated with a partial item

Unique identification is not possible if neither source_name and source_native_key within the sys_object_source_info object, nor the full set of identification criterion attributes specified for the CI class, is provided.

Incomplete items are saved as incomplete payloads in the CMDB IRE Incomplete Payloads [cmdb_ire_incomplete_payloads] table in JSON format. Incomplete items are stored for the purpose of logging payloads with irrecoverable errors, and are never processed again.

Adding relationships

Add relationships by using either indices, or the optional JSON internal_id element.

Use the relations object in the payload to add or update relationships by referring to internal_ids of items. Relationships can be created using main items and related items in the payload. For example:

- Relation (parent Index, child Index, Relation Type)
- Relation (parent Internal Id, child Internal Id, Relation Type)

For more information and for code samples, see [CreateOrUpdateCIEnhanced\(\)](#).

Adding references between payload items

Add references between two payload items by using the optional JSON internal_id element, which uniquely identifies payload items.

Use the referenceItems block to add or update references. You can add references between any two items, including main items, lookup items, and related items, in a single payload.

For more information and for code samples, see [CreateOrUpdateCIEnhanced\(\)](#).

CI reclassification

Use the updateWithoutUpgrade, updateWithoutDowngrade, and updateWithoutSwitch flags in the settings block in a payload, to prevent unintentional updates to CIs' class. These flags prevent upgrading, downgrading, or switching the class of a CI that multiple data sources unintentionally might attempt while updating the same CI. For more information and for code samples, see [CreateOrUpdateCIEnhanced\(\)](#).

Reclassification flags have precedence over any other system settings for CI reclassification during IRE processing.

Adding custom before and after scripts

Use the [IntegrationHub ETL](#) to [add custom Java scripts for a data source of a CMDB integration application](#). Those scripts provide access to the IRE input and output payloads, while processing CMDB integrations.

Before scripts provide access to a batch of input payloads that will be sent to IRE. Using a custom before script lets you:

- Skip a payload in the batch by setting the status to **SKIPPED**. Optionally, provide a reason for skipping the payload which will appear as a comment on the respective import set row table.
- Modify the input payload.
- Write other custom logic inside the script that uses the IRE payload.

After scripts provide access to the IRE input and output payloads. Using a custom after script lets you:

- Easily compare the input and output payloads and identify the different operations that IRE performed on each CI.
- Access the sys_ids of the CIs that IRE created or updated.
- Write other custom logic inside the script that uses the IRE output payload.

Apply CI Identification and Reconciliation to Import Sets

You can apply CMDB Identification and Reconciliation processes when Import Sets are used to import CIs into the CMDB. CI identification can prevent duplicate CIs in the CMDB, which Import Sets might otherwise cause.

Populating CMDB tables using Import Sets can inadvertently result in duplicate CIs when multiple imported records are identical to an existing CI. To minimize this duplication, you can apply CMDB Identification and Reconciliation processes to Import Sets when importing new records into CMDB tables.

Transform map script

In the onBefore transform map script for an import set, add a call to the [CMDBTransformUtil](#) API, similar to the following code sample:

```
(function runTransformScript(source, map, log, target) {
    // Call CMDB API to do Identification and Reconciliation o
    f current row
    var cmdbUtil = new CMDBTransformUtil();
    cmdbUtil.identifyAndReconcile(source, map, log);
    ignore = true;

    if (cmdbUtil.hasError()) {
        var errorMessage = cmdbUtil.getError();
        log.error(errorMessage);
    } else {
        log.info('IE Output Payload: ' + cmdbUtil.getOutputPayload());
        log.info('Imported CI: ' + cmdbUtil.getOutputRecordSysId());
    }
})(source, map, log, target);
```

The `ignore = true` code phrase prevents Import Sets from creating the same record again after it is processed by the identification engine.

Process

The identification engine performs identification of each source record before it is inserted into the CMDB. The identification engine determines if the record is a duplicate of an existing CI, and then:

- If not duplicate: Inserts the record to the target table.
- If duplicate: Updates the existing CI in the CMDB, with data from the source record.

The CMDBTransformUtil API pre-processes the source data, then passes the input values to the identification engine with import set being the data source by default. The CMDBTransformUtil API supports a target field that is a reference field in the same manner that Import Sets supports it. The CMDBTransformUtil API also supports a source script, evaluating source scripts to determine the target value which is then passed to the identification engine. For more information, see [Creating a field map](#).

Specify multiple target tables for an import set

You can configure each record in an import set with its own target table. Then, instead of inserting all the transformed records into a single target table, the records are inserted into the different target tables that are specified per record. For example, you might need to insert some records from the import set to the Computer class and other records to the Server class.

When [importing data using Import Sets](#), incorporate the following steps:

- In the data source file, add a target table column. Use a string such as "MyTable" to label the column header. In each record row, enter the target table for the record, as a valid CMDB class name such as "cmdb_ci_computer".
- After you **Auto Map Matching Fields** on the Table Transform Map form, add a field map for the added target table column to establish a relationship between classes and the target tables in the CMDB.
 1. In the **Field Map** related list on the Table Transform Map form, click **New**.
 2. Set **Source field** to the header of the target table column that you added in the data source file, such as **MyTable**.
 3. Set **Target field to Class**.
 4. Click **Submit**.

When you configure an import set with multiple target tables as described in the steps above, the **Target table** that is specified on the Table Transform Map form is not used.

Restrictions

The following restrictions apply:

- An import set should be associated with a single transform map. While adding a call to the CMDBTransformUtil API, ensure that still a single transform map exists for the import set.
- The CMDBTransformUtil API does not check if mandatory fields have values when used with Import Sets . Regardless of how enforce

mandatory fields is set in the transform map, data import fails if a mandatory field does not have a value.

- CI Identification and Reconciliation cannot be applied to Import Sets for dependent CIs (CIs with dependent identification rules).

Identification rules

The CMDB identification process relies on identification rules to uniquely identify CIs.

An identification rule applies to a CI class and consists of a single CI identifier and one or more identifier entries and related entries, each with a different priority. Each identifier entry defines a unique attribute set with a specific priority and each related entry defines rules for identifying related items. Create strong identification rules that are set with the highest priority for the strongest identifier entries and related entries.

The identification process and identification rules use the CIs attributes for identification:

Unique attributes

Designated sets of criterion attribute values of a CI, that can be used to uniquely identify the CI. Unique attributes can be from the same table or from derived tables.

Required attributes

Designated attributes of a CI that cannot be empty.

Derivation across the CMDB hierarchy

If no identification rule is explicitly defined for a child class, then the child class derives its identification rule, including any associated identification entries and related entries, from its parent class. Later, an own identification rule can be explicitly defined for the child class. In that case, the identification rule that was initially derived from the parent class, including any associated identification entries and related entries, is no longer in effect at the child class. Also, you must explicitly add identification entries and related entries in the newly created identification rule at the child class.

For example: The Hardware class identification rule has a related entry for the Software Instance table. This identification rule, including its associated related entry for the Software Instance table, is derived by the Computer class. If you then create a new identification rule for the Computer class, it overwrites the identification rule that was derived from the Hardware class. Therefore, the Hardware class identification rule, including its associated related entry for the Software Instance table, is no longer in effect for the Computer class. If the same related entry is needed, you must explicitly add a related entry for the Software Instance table in the newly created identification rule for the Computer class.

Identification rule types

CI dependency is specified in the CI Class Manager by the dependent relationship rules for the CI's class:

Independent CIs

CIs, such as Server CIs, which exist on their own and are not dependent on any other CIs.

Dependent CIs

CIs which depend on a relationship to another CI and can't exist on their own in the absence of the dependent relationship. For example:

- Network Adapter CIs can't exist meaningfully without the Hardware CIs that contain them.
- Application CIs can't exist on their own without the Server CI they are hosted on.

The steps for identifying dependent CIs can be different from the steps for identifying independent CIs. This difference is reflected in the differences between dependent identification rules and independent identification rules:

Independent identification rule

A rule that identifies a CI based on the CI's own attributes, independently of other CIs or relationship.

Dependent identification rule

A rule in which identifying a CI requires identifying a dependent CI first. A CI can have dependency on one or more CIs, and a dependent CI can have only a single parent CI with dependency. The relationship types between the CI and its dependent CIs are also included in the identification process. To help with the identification process of dependent CIs, [create dependent relationships](#) that define the dependency chain within CI types.

The payload used for identification of a dependent CI, can include a relationship with a qualifier chain. For such relationship, if there is a matching parent/child pair, the system compares the qualifier chain in the payload, with the qualifier chain of the CIs in the database. If there is a difference, the qualifier chain in the database is updated to match the qualifier chain in the payload for that relationship.

Identifier entries

You can configure an identifier entry to match a CI not only based on the CI's own attributes (field-based identification) but also based on the CI's related list (lookup-based identification) such as **Serial Numbers** or **Network Adapters**. The lookup table that is used for identification, needs to have a reference field that points to cmdb_ci.

There are three types of identifier entries:

Regular identifier entry

Based on CI's attributes that uniquely identify the CI.

Lookup identifier entry

Uses a lookup table (related table) which can be any table that has a reference to the CI that is being identified. After you select a related lookup table, you select identifier attributes from the related table that reference either the cmdb_ci table itself, or one of its descendants.

If the lookup records do not already exist, then they are inserted in the lookup table referenced in the identifier entry.

Hybrid identifier entry

A combination of both, a regular identifier entry and a lookup identifier entry.

Example: When discovering virtual machines in a cloud environment which might contain two virtual machines with an identical set of serial numbers. A lookup identifier entry for the Hardware table such as [Table: Serial Number, Criterion Attributes: Serial Number, Serial Number Type] cannot uniquely identify these two virtual machines. However, a hybrid identifier entry such as [Table: Serial Number, Criterion Attributes: Serial Number, Serial Number Type + (Name field from main Hardware table)] can uniquely identify the two virtual machines.

Guidelines for lookup tables

Follow these guidelines when specifying a lookup table in an identifier entry.

1. Ensure that lookup tables reference the cmdb_ci table.
2. It is preferable to enforce exact count match (check box **Enforce exact count match (Lookup)**) for a stronger identification rule. During lookup identification, this option enforces matching only on exact lookup records count match. See [Create or edit a CI identification rule](#) for more details.
3. Do not create conflicting identification rules especially for lookup-based rule.

Example: In a CI Identifier for the Hardware class, you specify a lookup-based rule for the Network Adapter class and you also define a CI Identifier for the Network Adapter class. Duplicates might potentially be created in the Network Adapter table, because there are contradicting rules to identify a unique CI in that table:

- One rule that looks only at criterion attributes (CI identifier rule)
- Another rule that looks at criterion attributes and referenced sys_id (lookup rule).

Example: CI with related items that needs to be inserted - sysId is available.

```
var payload = {
  items: [
    {
      className:'cmdb_ci_linux_server',
      related: [
        {
          className:'cmdb_ci_spkg',
          values: {
```

```
        name:'package1',
        version:'version1'
    }
},
values: {
    sys_id:'194876usytrr65378098'
}
];
};
```

Related entries

You can define related entries which are rules that are based on related Cls. A related entry is based on a related table which can be any table (CMDB or non-CMDB) that has a reference to the CI that is being identified. Related entries let you create or update records on other tables in which the data is associated with the CI being identified by the identifier entries. Related entries are not used to directly identify Cls.

After you select a related table for the rule, the list in **Referenced field** is populated with fields from the related table that reference either the cmdb_ci table itself, or one of its descendants.

A related entry for a class is derived by child classes for which no related entries are specified.

- [Create or edit a CI identification rule](#)

Identification rules are used to uniquely identify Cls in the CMDB, as part of the identification and reconciliation process. Each CMDB class can be associated with a single identification rule.

- [Create an identification inclusion rule](#)

Narrow the scope of Cls that are included in the identification process by creating an identification inclusion rule.

Related concepts

- [Relation qualifier](#)

Identification rules are used to uniquely identify Cls in the CMDB, as part of the identification and reconciliation process. Each CMDB class can be associated with a single identification rule.

Before you begin

Role required: itil has read access, itil_admin (on top of itil) has full access.

About this task

In a CI identification rule, specify a CI identifier, and identifier entries and related entries that uniquely identify the CI.

Review the following before creating identification rules:

- Identification rules
- Effective usage of CMDB Identification

Procedure

1. Navigate to **All > Configuration > CI Class Manager**.
2. Click **Hierarchy** to display the CI Classes list. Select the class for which to create an identification rule.
3. In the class navigation bar, expand **Class Info** and then click **Identification Rule**.
4. Click **Edit** to edit an existing rule, or click **Add** in the Identification Rule section to create one. Fill out the form, and then click **Save**.

Field	Description
Independent/Dependent	Designation of whether the CI identifier can identify the CI independently of other CIs, or not. Note: To set the rule as Dependent , you must specify dependent relationship rules for the selected class.
Name	Name of CI identifier.

Field	Description
Description	Description of the CI identifier.

5. In the Identifier Entries section, click an existing identifier entry to edit, or click **Add** to create one.
6. In the Identifier Entry dialog box, choose an option and then click **Next**. Continue with one of the following three steps according to the option you selected.

Option	Description
Use attributes from main table <table>	Lets you select attributes from the currently selected table (regular identifier entry).
Use attributes from another table (Lookup table)	Lets you select attributes from any related table, other than the currently selected table (lookup identifier entry).
Use attributes from main and another table (Hybrid)	Lets you select attributes from both the currently selected table, and from another table (hybrid identifier entry).

7. **Use attributes from main table <table>** option: Set the options on the form and then click **Save**.

Search On Table is preset to the currently selected table in the CI Classes list.

Field	Description
Active	Check box that specifies the identifier entry is active. At least one identifier entry in an identification rule must be active for the rule to apply.

Field	Description
Priority	<p>Priority of the identifier entry. Identifier entries are applied based on priority. Rules with lower priority numbers are given higher priority. Identifier entries of identical priorities are applied randomly.</p> <p>You can keep gaps between the priority numbers, so you can assign the unused priority numbers to new entries without modifying the existing priority order.</p>
Criterion Attributes	<p>Set of attributes that uniquely identify the CI. Attributes can belong to the current class, or to a parent class.</p>

Field	Description
	<p>Note: It is possible to add reference fields as a criterion attribute. However, such fields might not always be effective:</p> <ul style="list-style-type: none"> Reference fields store sys_ids that point to a record in another table, and thus is considered a weak criterion attribute (in terms of uniqueness) for the current table. The system detects and then replaces invalid values in a reference field with 'Unknown'. For example, an invalid Model ID value is replaced with the value 'Unknown'. Also, if several CIs end up having that same reference field set to 'Unknown', then these CIs become duplicate CIs.
Allow null attribute	<p>When selected, then if at least one criterion attribute is not null, attempt matching with an identifier entry even if there are criterion attributes that are null.</p> <p>Otherwise, all criterion attributes must have values to attempt matching with an identifier entry.</p>

Field	Description
Allow fallback to parent's rules	Allows the identification rules of the CI's parent to be used if a match is not found for this identification rule. Applies only for dependent identification rules.
Advanced Options	<p>A filter to narrow the set of records that will be searched for a matching CI.</p> <p>Available only if the <code>glide.identification_engine.enable_identifier_optional_condition</code> system property is set to true (false by default). In the base system, identifier entries of various classes are pre-configured with advanced options conditions. All these pre-configured conditions in regular identifier entries will automatically apply when you set this property to true. Therefore, to prevent unexpected behavior, review those predefined conditions in regular identifier entries before setting this property to true.</p> <p>For more details about this property, see Properties for Identification and Reconciliation.</p>

Note: If criterion attributes have only two attributes and sys_class_name is one of them (for example [name, sys_class_name], [ip_address, sys_class_name]), then the other attribute cannot be NULL, even if **Allow null attribute** is enabled. This restriction is due to sys_class_name being considered a special system matching attribute.

8. **Use attributes from another table (Lookup table)** option:

- a. Set **Search On Table** to a table other than the currently selected table in the CI Classes list. The **Search On Table** must have a reference field to cmdb_ci, otherwise the identifier entry is considered invalid.
- b. Set the rest of the fields as described in the previous step.
- c. (Optional) Click **Advanced options** and enter the information for a lookup identifier (scroll down if necessary).

Advanced Option	Description
All of these conditions must be met	A filter to narrow the set of records that will be searched for a matching CI.
Enforce exact count match	<p>For lookup identification, match a CI only on exact lookup records count match. When enforced, all lookup items for a CI in the payload must have matching records in the lookup table, that reference the same CI:</p> <ol style="list-style-type: none">a. Only matches CIs that have all the lookup items from the input payload referencing the CI in CMDB.b. If there are multiple matches, selects the oldest

Advanced Option	Description
	<p>created CI as the final match.</p> <p>When not enforced, one lookup item for a CI in the payload matching a record in the lookup table, is sufficient to consider a match:</p> <ol style="list-style-type: none"><li data-bbox="878 756 1279 914">Matches any CI that has at least one of the lookup items from the input payload referencing the CI in CMDB.<li data-bbox="878 946 1279 1125">If there are multiple matches, selects the CIs with the max number of lookup items from the input payload referencing the CI in CMDB.<li data-bbox="878 1157 1279 1284">If there are still multiple matches, selects the oldest created CI as the final match.

- Click **Save**.

9. **Use attributes from main and another table (Hybrid)** option:

- Set the options on the **General Settings** tab as described in previous steps, and then click **Next**.
- On the **Main Table Settings** tab, select the attributes to use from the currently selected table, and then click **Next**.
Search On Table is preset to the currently selected table in the CI Classes list.
- On the **Lookup Table Settings** tab, select a **Search On Table** and then in **Criterion Attributes** select attributes from the specified table. **Search On Table** must have a reference field to cmdb_ci, otherwise the identifier entry is considered invalid.

You can click **Advanced options** and enter the information for a lookup identifier as described in the previous step (scroll down if necessary).

- d. Click **Save**.

Note: The **Allow null attribute** option in the hybrid option, is set to **false**. Therefore, all of the selected criterion attributes from both the currently selected table and the lookup table, must have a value. Also, setting optional conditions is available only for the lookup table, and is not available for the main table.

10. (Optional) On the Related Entries section click an existing related entry to edit, or click **Add** to create one.

- a. Update the Related Entry form and then click **Save**.

Related Entry form

Field	Description
Active	Check box that specifies that the related entry is active.
Related table	A related table that references the CI that is being matched.
Referenced field	A referenced field in Related table that should store the referenced CI. This field always references the cmdb_ci table, or a descendent of the cmdb_ci table.
Priority	Priority of the related entry for the specified Related table . Rules with lower priority numbers are given higher priority while matching a related item for specific related table. Related entries for the specified related table with identical priorities are applied randomly.

Field	Description
	You can keep gaps between the priority numbers, so you can assign the unused priority numbers to new entries without modifying the existing priority order.
Criterion attributes	The set of attributes to uniquely identify the related item. Attributes can belong to the current class, or to a parent class.

Field	Description
	<p>Note: It is possible to add reference fields as a criterion attribute. However, such fields might not always be effective:</p> <ul style="list-style-type: none"> Reference fields store sys_ids that point to a record in another table, and thus is considered a weak criterion attribute (in terms of uniqueness) for the current table. The system detects and then replaces invalid values in a reference field with 'Unknown'. For example, an invalid Model ID value is replaced with the value 'Unknown'. Also, if several Cls end up having that same reference field set to 'Unknown', then these Cls become duplicate Cls. <p>Click the lock icon to view, add, or remove attributes from the identification rule.</p>
Allow null attribute	If at least one criterion attribute in the related table is not null, allow to attempt matching with an identifier

Field	Description
	entry even if there are criterion attributes which are null.
Filter conditions	Add conditions to construct a filter to narrow the set of records that will be searched for a matching related item.

Note: If criterion attributes have only two attributes and sys_class_name is one of them (for example [name, sys_class_name], [ip_address, sys_class_name]), then the other attribute cannot be NULL, even if **Allow null attribute** is enabled. This restriction is due to sys_class_name being considered a special system matching attribute.

Example

For example, the pre-defined **Hardware Rule** applies to the Hardware [cmdb_ci_hardware] table. It has an identifier entry with the criterion attribute **Serial Number**, **Serial Number Type** and its **Search on table** field is set to **Serial Number**.

The following payload snippet adds a CI to the cmdb_ci_linux_server class, that is a child of the Hardware class. It also shows how you can add related items in the payload for which you should create **Related Entries** on the CI Identifier page for the Hardware [cmdb_ci_hardware] table:

```
{  
  "items": [  
    {  
      "className": "cmdb_ci_linux_server",  
      "lookup": [  
        {  
          "className": "cmdb_serial_number",  
          "values": {  
            "serial_number": "VMware-42 21 e  
3 da 44 14 5a a6-56 48 2b 0a 28 53 42 4c",  
            "serial_number_type": "system",  
            "valid": "true"  
          }  
        }  
      ]  
    }  
  ]  
}
```

```
        },
        {
            "className": "cmdb_serial_number",
            "values": {
                "serial_number": "4221E3DA-4414-5
AA6-5648-2B0A2853424C",
                "serial_number_type": "uuid",
                "valid": "true"
            }
        },
    ],
    "related": [
        {
            "className": "cmdb_ci_ucs_chassis",
            "values": {
                "name": "chassis1",
                "category": "category1",
                "short_description": "My Chassis
1"
            }
        },
        {
            "className": "cmdb_ci_ucs_chassis",
            "values": {
                "name": "chassis2",
                "category": "category2",
                "short_description": "My Chassis
2"
            }
        },
    ],
    "values": {
        .....
        "name": "xpolog2.lab3",
        "os_name": "Linux",
        "output": "Linux xpolog2.lab3 2.6.32-431.
el6.x86_64 #1 SMP Fri Nov 22 03:15:09 UTC 2013 x86_64 x86
_64 x86_64 GNU/Linux",
        "serial_number": "VMware-42 21 e3 da 44 1
4 5a a6-56 48 2b 0a 28 53 42 4c",
        "sys_class_name": "cmdb_ci_linux_server"
    }
}
```

```
}
```

When the **Hardware Rule** is applied, the Serial Number [cmdb_serial_number] table is searched for a match with the values specified within the lookup key. Unless **Enforce exact count match (Lookup)** is checked, it is not necessary for every lookup key to return a match, as long as there is at least one match. If all matches reference the same CI, then that CI is considered to be the existing CI record. If no match is found, then the identification search continues to the next rule entry. If after all the rules are exhausted without finding a match, a new CI record is created in the database.

What to do next

You can optionally [create an inclusion rule](#) to narrow the scope of CIs that are included in identification.

Related tasks

- [Create an identification inclusion rule](#)

Narrow the scope of CIs that are included in the identification process by creating an identification inclusion rule.

Before you begin

Role required: itil has read access, itil_admin (on top of itil) has full access.

About this task

During duplication detection of independent CIs, the identification and reconciliation engine (IRE) processes only the CIs that satisfy the identification inclusion rules. For example, you can set a filter to include only CIs whose state is operational. When no identification inclusion rules exist, all CIs are included in the identification process and in the CMDB Health duplicate metric calculations. In the base system, there are no predefined identification inclusion rules. Identification inclusion rules are defined at the class level.

Identification inclusion rules also indirectly impact what appears in CMDB health dashboards for duplicate CIs, in addition to any [health inclusion rules](#).

Note: Identification inclusion rules impact any script that calls IRE, therefore create them carefully. Identification inclusion rules can prevent the identification of certain types of CIs, affecting some features of Discovery and Service Mapping.

Procedure

1. Navigate to **All > Configuration > CI Class Manager**.
2. Click **Hierarchy** to display the CI Classes list. Select the class for which to create an identification inclusion rule.
3. In the class navigation bar, expand **Class Info** and then click **Identification**.
4. In the Inclusion Rule (Advanced) section, click **Add** to create a rule or click **Edit** to edit an existing rule. In the Create Inclusion Rules dialog box, specify a criteria in the **Active record condition** field. CIs must meet this criteria to be included in the identification process and in the duplicate CMDB Health metric.
5. Click **Save**.

What to do next

Navigate to **All > Configuration > Identification/Reconciliation > Identification Inclusion Rules** to see the list of all identification inclusion rules.

Related tasks

- [Create or edit a CI identification rule](#)
- [Create health inclusion rule](#)

Reconciliation rules

Reconciliation rules determine which discovery sources can update CI attributes.

Discovery sources, such as EventManagement, ImportSet, ManualEntry, and Tivoli, are used with the [createOrUpdateCI\(\)](#) API to simulate manual updates to CIs. Without reconciliation rules, discovery sources can overwrite each other's updates to attribute values.

There are two types of reconciliation rules:

Static reconciliation rules

Static reconciliation rules are the legacy reconciliation rules that set priorities for the various discovery sources for updating CI attributes. Static reconciliation rules specify which discovery sources can update class attributes, and the precedence order among these discovery sources.

When creating static reconciliation rules, ensure that there is a reconciliation rule for each discovery source that is authorized to update an attribute. Reconciliation rules can be defined at the parent and the child class level.

Static reconciliation rules are stored in the Reconciliation Definition [cmdb_reconciliation_definition] table.

Dynamic reconciliation rules

Dynamic reconciliation rules are based on attribute values processed by [CMDB 360/Multisource CMDB](#) rather than on discovery source priority. First, CMDB 360 processes the current payload data into the CMDB 360 data store. Then, applying a dynamic reconciliation rule, IRE selects the largest or most reported value, for example, across all discovery sources. Because dynamic reconciliation rules leverage CMDB 360, you must enable that feature to use dynamic reconciliation rules.

Creating dynamic reconciliation rules can be useful, for example, if it becomes difficult to set priority order for multiple discovery sources. Only a single dynamic reconciliation rule can exist per class attribute.

Dynamic reconciliation rules are stored in the Dynamic Reconciliation Definitions [cmdb_dynamic_reconciliation_definition] table.

Examples of static reconciliation rules

The following sample static reconciliation rules are created for the cmdb_ci_computer class and its cmdb_ci_linux_server child class:

1. Discovery is exclusively authorized to update the name attribute in the cmdb_ci_computer class.

Because reconciliation rules are derived by child classes from parent classes, this rule also authorizes Discovery to update the name attribute in any child classes for the cmdb_ci_computer class.

2. ServiceWatch is exclusively authorized to update the name attribute in the cmdb_ci_linux_server class.
3. ServiceWatch is exclusively authorized to update all attributes in the cmdb_ci_linux_server class, as configured by leaving the **Attributes** field empty in the rule.

See [Create a CI reconciliation rule](#) for details about creating a static reconciliation rule that, for example, authorizes a discovery source to update a specific attribute such as name.

Using reconciliation rules

As you create reconciliation rules, keep in mind the following principles which are designed for flexibility and the refinement of rules at the attributes level:

Precedence of dynamic reconciliation rules

When both, static and dynamic reconciliation rules exist for the same CI attribute, the dynamic reconciliation rule takes precedence over the static reconciliation rule.

Authorization for all attributes in a class

A static reconciliation rule lets you authorize a discovery source to update all attributes in a class. However, this authorization can be overridden for some of the attributes by rules for child classes in which specific attributes are listed.

For example, if only example rules #1 and #3 above are created, then Discovery is authorized to update the name attribute in the

cmdb_ci_linux_server class. ServiceWatch is authorized to update all other attributes in the class except for the name attribute.

To override the authorization of Discovery to update the name attribute, example rule #2 above is added to specifically authorize ServiceWatch to update the attribute.

Authorization to only specific attributes in a class

To authorize a discovery source to update specific attributes in a class, create a static reconciliation rule for the discovery source, and list these attributes in the rule. A rule that grants access to specific attributes in a class overrides other static reconciliation rules with an empty attribute list that grants access to the entire class.

Example rule #1 above grants Discovery with exclusive authority to update the name attribute of the cmdb_ci_computer class. All other discovery sources are prevented from updating the name attribute of any CI in the cmdb_ci_computer class.

Child class rules overrides parent class rules

Any reconciliation rules defined for a child class override the rules defined for its parent class. This rule applies also when the child's reconciliation rule is static and the parent's rule is dynamic (dynamic reconciliation rules have precedence over static reconciliation rules when they are for same level class).

For example, rule #1 above lets Discovery update the name attribute in the cmdb_ci_computer class and all of its child classes. However, rule #2 for the cmdb_ci_linux_server child class, which overrides rule #1 for the parent class, explicitly authorizes ServiceWatch to update this attribute in the child class.

As a result:

- Discovery cannot update the name attribute of the child cmdb_ci_linux_server class. Only ServiceWatch is authorized to update this attribute.
- Discovery is authorized to update the name attribute of CI records in all other child classes of the cmdb_ci_computer class.

Overlapping static reconciliation rules

Static reconciliation rules that authorize different discovery sources for the same attributes of the same class can coexist and do not exclude each other.

For example, assume the following rule is added. It is similar to example rule #1 above but authorizes a different discovery source:

ServiceWatch is authorized to update the name attribute in the cmdb_ci_computer class.

Like example rule #1 above, this new rule applies to the name attribute in the cmdb_ci_computer class so both Discovery and ServiceWatch can update the attribute. Any reconciliation rules are enforced to prevent the discovery sources from overwriting each other's updates.

For more information about reconciliation rules, see the [\[CMDB - Data Precedence Rules\] Understanding the CMDB data precedence rules and troubleshooting \[KB0756709\]](#) knowledge base article (Starting with the Paris release, reconciliation and data precedence rules are merged).

Domain separation

If Domain Separation is enabled, then you can scope reconciliation rules to specific domains. Rules of the parent domain, if not overridden, apply to CIs of child domain. All rules that are visible to a domain are applied, and a rule overriding the parent domain displays the child domain version.

[Understanding the CMDB reconciliation rules and troubleshooting \[KB0756709\]](#)

- [Create a CI reconciliation rule](#)

Create a static or a dynamic CI reconciliation rule.

- [Create a data refresh rule](#)

Specify data refresh rules to determine if a CI is stale for a specific discovery source. Such CIs can then be updated by a lower-priority authorized discovery source.

Create a static or a dynamic CI reconciliation rule.

If both, static and dynamic reconciliation rules exist for the same CI attribute, the dynamic rule has precedence.

Note: You can't create a reconciliation rule for system fields or for Identification and Reconciliation Engine (IRE) specific fields such as the Discovery source (discovery_source) field. Also, reconciliation rules can't be dot-walked using reference fields.

Related tasks

- [Create a data refresh rule](#)

Create a static reconciliation rule

A static reconciliation rule specifies class attributes that discovery sources are authorized to update, and prevents unauthorized discovery sources from overwriting the attributes' values. A static reconciliation rule also specifies the prioritization among multiple discovery sources. Without static reconciliation rules, discovery sources can overwrite each other's updates to attribute values.

Before you begin

Role required: itil has read access, itil_admin (on top of itil) has full access.

About this task

Static reconciliation rules are used in conjunction with [data refresh rules](#) to determine reconciliation steps for a CI. These rules determine if, when, and by which discovery source a CI can be updated. If multiple discovery sources are authorized to update the same class attributes, assign a priority to each of these discovery sources to prevent them from overwriting each other's updates.

After an authorized discovery source updates an attribute, subsequent updates are accepted only from the same discovery source or from a discovery source with a higher priority. Updates from a discovery source with a lower priority are rejected, unless these two conditions are met:

- The lower priority source is the first source updating the CI.
- The CI became stale based on data refresh rules for the CI class.

Precedence order of static reconciliation rules:

- Rule configured for a specific attribute, has precedence over rule set with **Apply to all attributes** (regardless of priority value).
- Between two rules for the same attribute or between two rules set with **Apply to all attributes**, the rule that is specific directly for the class has precedence over the derived rule.
- Between two rules for the same attribute or between two rules set with **Apply to all attributes** at the same class level, precedence is determined by rule priorities.

Information about the last discovery source that updates each attribute is stored in the Data Source History [cmdb_datasource_last_update] table, but only after enabling the reconciliation rule. Therefore, there might be unexpected updates after you enable the rule until the highest priority data source has updated the CI.

Static reconciliation rules affect reconciliation of stale CI attributes. During reconciliation, the information in the Data Source History table is considered along with the data refresh rules for the CI's class, to determine if a CI attribute is stale. A CI attribute is determined to be stale if it was not updated by the latest discovery source to update the CI, within a time period. The time period is specified by the Effective Duration time in the data refresh rule for the class for the discovery source. In this case, if another authorized discovery source, with a lower priority attempts to update the stale CI attribute, the update is allowed.

If there is a dynamic reconciliation rule for the same CI attribute as in a static reconciliation rule, the dynamic rule takes precedence.

Procedure

1. Navigate to **All > Configuration > CI Class Manager**.
2. Click **Hierarchy** to open the CI Classes hierarchy list. Then select a class for which to create a reconciliation rule.
3. In the class navigation bar, expand **Class Info** and then click **Reconciliation Rules**.
4. In the Reconciliation Rules section, click **Add** to create a rule or select an existing rule to edit.

5. Click the **Static Reconciliation Rule** tile if it appears.
If **CMDB 360/Multisource CMDB** is not enabled, you can't create a dynamic reconciliation rule and the tiles to choose the rule type do not appear.
6. Fill out the fields on the **Add Data Sources & Prioritize** tab, and then click **Next**.

Field	Description
Active	Check box to activate this reconciliation rule.
Discovery Source	The discovery source that you are configuring this rule for.
Priority	Priority of Source within other discovery sources for the specified attributes. Smaller numbers designate higher priority. Discovery sources without a reconciliation rule are assigned the lowest priority.

You can add multiple pairs of **Discovery Source** and **Priority**.

7. Fill out the fields on the **Select Attributes** tab, and then click **Next**.

Field	Description
Apply to all attributes	<p>Authorizes the specified discovery sources to update all attributes of the specified class.</p> <p>Note: This rule will be overridden by any rule that applies to a specific attribute. In which case, instead of using this option, you can directly include all attributes for Attributes.</p>

Field	Description
Attributes	<p>Attributes, from the current or from a parent class, that the specified discovery sources are authorized to update.</p> <p>Available only if Apply to all attributes is not selected.</p>
Update with Null	<p>Attributes that the specified discovery sources can update with a null value. By default, authorized discovery sources cannot overwrite a non-null value with a null value.</p> <p>Attributes in this list, which are not in the Attributes list, are not included with the attributes that the discovery sources can update with a null value.</p>

8. Fill out the fields on the **Set Filter Condition** tab, and then click **Save**.

Field	Description
Filter Condition	<p>Conditions that CIs must meet for the rule to be applicable.</p> <p>For example, to apply this rule only to CIs that are associated with the Finance department, select this condition: [Department] [is] [Finance].</p>

Note: The `glide.identification_engine.enable_reconciliation_filter_before_update` system property determines when filter conditions are applied. By default, those filter conditions are applied after attribute values have changed during payload processing. Set this property to **true** so that Identification and Reconciliation Engine (IRE) applies the filter conditions before attribute values change.

What to do next

- Click the filter icon () and select:
 - **Attributes:** To show only reconciliation rules for a specific attribute.
 - **Discovery sources:** To show only reconciliation rules for a specific discovery source.
- Click **Preview Rule** to see per attribute, the precedence order between any discovery sources that are authorized to update that attribute and any dynamic reconciliation rules.
- If CMDB 360/Multisource CMDB is enabled, you can:
 - Click **Preview Data** to see all attributes for a specific CI. Also, for each attribute, the current CMDB value and discovery sources reported values for the attribute.
 - Click **Recompute** to **recompute CI attribute values** after changing reconciliation rules.
- Navigate to **All > Configuration > Identification/Reconciliation > Reconciliation Definitions** to see a list view of all definitions of reconciliation rules.

Create a dynamic reconciliation rule

A dynamic reconciliation rule uses CMDB 360 data to choose a value such as the largest value that is reported, for updating a CI.

Before you begin

CMDB 360/Multisource CMDB must be enabled.

Role required: itil has read access, itil_admin (on top of itil) has full access

About this task

If the same CI attribute has both, a static reconciliation rule and a dynamic reconciliation rule, the dynamic reconciliation rule has precedence.

A dynamic reconciliation rule supports several rule types, such as largest reported value and most reported value. When applying a dynamic reconciliation rule, IRE processes the current payload and then examines the CMDB 360 data store to select a value with which to update the CMDB. Depending on the dynamic reconciliation rule type, selecting the appropriate value might not be immediately conclusive. For example, there might not be a single value that is most reported, or for some values, the last discovered timestamp isn't reported. Therefore, when necessary, IRE falls back to examining additional details such as last reported, last discovered, and last updated values to select the most appropriate value.

Note: You can't add a dynamic reconciliation rule when creating a new child class in the CI Class Manager. You must first save the new child class and then add the dynamic reconciliation rule.

Procedure

1. Navigate to **All > Configuration > CI Class Manager**.
2. Click **Hierarchy** to open the CI Classes hierarchy list. Then select a class for which to create a reconciliation rule.
3. In the class navigation bar, expand **Class Info** and then click **Reconciliation Rules**.
4. In the Reconciliation Rules section, click **Add** to create a rule or select an existing rule to edit.
5. Click the **Dynamic Reconciliation Rule** tile.
6. On the **Select Rule** tab, select a rule type in the **Dynamic Rule Type** list field, and then click **Next**.

7. On the **Select Attributes** tab, select the attributes for which to apply the rule. Then click **Next**.

Attributes that the specified rule type can't be applied to and attributes for which a dynamic reconciliation rule already exists for, don't appear.

8. Fill out the fields on the **Set Filter Condition** tab, and then click **Save**.

Field	Description
Filter Condition	Conditions that CIs must meet for the rule to be applicable. For example, to apply a rule only to CIs that are associated with the Finance department, select this condition: [Department] [is] [Finance] .

What to do next

- Click the filter icon () and select:
 - **Attributes:** To show only reconciliation rules for a specific attribute.
 - **Discovery sources:** To show only reconciliation rules for a specific discovery source.
- Click **Preview Rule** to see per attribute, the precedence order between any discovery sources that are authorized to update that attribute and any dynamic reconciliation rules.
- Click **Preview Data** to see all attributes for a specific CI. Also, for each attribute, the current CMDB value and discovery sources reported values for the attribute.
- Recompute CI attribute values.

- Navigate to **All > Configuration > Identification/Reconciliation > Reconciliation Definitions** to see a list view of all definitions of reconciliation rules.

Specify data refresh rules to determine if a CI is stale for a specific discovery source. Such CIs can then be updated by a lower-priority authorized discovery source.

Before you begin

Role required: itil has read access, itil_admin (on top of itil) has full access.

About this task

Data refresh rules have no impact when dynamic reconciliation rules are in effect.

Data refresh rules are used in conjunction with static reconciliation rules to determine reconciliation steps for a CI. These rules determine if, when, and by which discovery source a CI can be updated.

Procedure

1. Navigate to **All > Configuration > CI Class Manager**.
2. Click **Hierarchy** to display the CI Classes list. Select the class for which to create a data refresh rule.
3. In the class navigation bar, expand **Class Info** and then click **Reconciliation Rules**.
4. In the Data Refresh Rules section, click **Add** to create a rule or select an existing rule to edit. Fill out the details in the Create Data Refresh Rules dialog box.

Field	Description
Discovery source	Discovery source for which staleness is evaluated.
Effective Duration	The time period that is used for the staleness test.

Field	Description
	If the fields specified in the static reconciliation rule for the CI's class were not updated by the specified discovery source within the specified time period — the CI is determined to be stale for that discovery source. If you enter a value with a prefix that is valid and a suffix that is not, such as 15 x — the valid portion of the value is used ('15'). If the entire value is invalid — the default value of 0 is used.
Active	Activates the rule.

5. Click **Save**.

Related concepts

- [Create a CI reconciliation rule](#)

Create an IRE data source rule

When using Identification and Reconciliation Engine (IRE), you can prevent a specific discovery (data) source from inserting new CIs for a specific class. Create IRE data source rules for discovery sources that you don't trust in creating CIs but continue to trust in updating those CIs that exist.

Before you begin

Role required: itil_admin

About this task

IRE data source rules have no impact when dynamic reconciliation rules are in effect.

For example, an IP scan tool that discovers network gear but does not discover servers and therefore creates server CIs without details. You can prevent such discovery source from creating specific CIs, while still permitting it to update those specific CIs if they exist. IRE data source rules are stored in the IRE Data Source Rule [cmdb_ire_data_source_rule] table.

- Child classes derive IRE data source rules from parent classes like identification rules do.
- IRE data source rules that are specified for a child class, override any IRE data source rules derived from a parent class.

When IRE processes an insert operation that is prohibited by an IRE data source rule, the insert operation fails. This failure happens when the discovery source and CI class in the insert operation and in an IRE data source rule, match. When [CreateOrUpdateCIEnhanced\(\)](#) is used, IRE stores the failed payload in the CMDB IRE Partial Payloads [cmdb_ire_partial_payloads] table for future potential use.

Note: When an insert operation is not allowed by the IRE data source rule, then when using [createOrUpdateCI\(\)](#), the entire IRE payload fails since [createOrUpdateCI\(\)](#) doesn't allow partial commits.

If later, a permitted discovery source attempts to insert that same CI, then IRE inserts the CI after merging it with the matching CI from the partial payloads. IRE then deletes the partial payload from the CMDB IRE Partial Payloads [cmdb_ire_partial_payloads] table, and allows future updates by the discovery source specified in the rule.

IRE data source rules do not apply to lookup and related items, and only a single rule can be active for any class/discovery source pair.

Procedure

1. Navigate to **All > Configuration > Identification/Reconciliation > IRE Data Source Rule**.
2. In the list view, click **New** and fill out the IRE Data Source Rule form.

Field	Description
Active	Activates the IRE data source rule.
Applies to	The class (and child classes) that the specified discovery (data) source is not allowed to create CIs of.
Data source	Discovery (data) source that is not allowed to create CIs of the specified class.
Insert Not Allowed	Disables the specified discovery (data) source from inserting new CIs from the specified class, to the CMDB.

3. Click **Submit**.

Result

If a payload item with an insert request, and in which the discovery source and the CI class match the discovery source and the CI class specified in the IRE data source rule:

1. The insert operation fails and IRE logs the following message:

INSERT_NOT_ALLOWED_FOR_SOURCE Insert into [xyz] is blocked for data source [xyz] by IRE data source rule.

2. If using [CreateOrUpdateCIEnhanced\(\)](#), then IRE stores the payload item as a partial payload in the CMDB IRE Partial Payloads [cmdb_ire_partial_payloads] table.

If later, a permitted discovery source successfully inserts a CI that matches the CI from a partial payload item:

1. The current CI is merged with the matching CI from the partial payload, applying static reconciliation rules as needed.

2. The respective partial payload in the CMDB IRE Partial Payloads [cmdb_ire_partial_payloads] table is deleted.
3. Later payloads in which the non-permitted discovery source updates the respective CI, run successfully.
4. IRE allows the discovery source, that was previously prohibited from inserting the CI, to update that same CI which now exists in the CMDB.

Detecting duplicate CIs

When the identification process encounters duplicate CIs, it groups each set of duplicate CIs into a de-duplication task for review and remediation. A large number of duplicate CIs might be due to weak identification rules. You can configure the identification engine to reconcile duplicate CIs.

During CMDB Identification, processing of duplicate CIs is determined by the properties `glide.identification_engine.skip_duplicates` (set to true by default) and `glide.identification_engine.skip_duplicates.threshold` (set to 5 by default), and on the number of duplicate CIs that are detected. You can configure these properties so duplicate CIs are automatically reconciled, skipping duplication.

- If `glide.identification_engine.skip_duplicates` is true, and the number of duplicate CIs is less than the threshold specified by `glide.identification_engine.skip_duplicates.threshold`, then the oldest of the duplicate CIs is picked as a match and gets updated. That oldest duplicate CI also becomes the main CI for that set of duplicate CIs. The rest of the duplicate CIs are tagged as duplicates by setting the `cmdb_ci`'s `duplicate_of` to the appropriate main CI. During matching, the identification engine filters out any CI that is tagged as duplicate of any CI.
- If `glide.identification_engine.skip_duplicates` is false, then matching of duplicate CIs fails with an error, and none of the duplicate CIs are updated.

In either case, de-duplication tasks are always created.

Note: For a duplicate CI, if any of the CI's attributes, other than `duplicate_of`, is updated by IRE processing, then the CI is no longer considered a duplicate CI. In that situation, the value of `duplicate_of` is cleared in the CI.

For more information about these properties, see [Properties for Identification and Reconciliation](#).

Review de-duplication tasks

For information about reviewing and remediating de-duplicate tasks, and how the main CI is used, see [Duplicate CIs](#).

Generate and simulate payload execution using identification simulation

Identification simulation is a central location for automatically constructing a payload that is guaranteed to be complete and valid. You can then simulate the processing of the payload by the identification and reconciliation engine (IRE) and examine the results before actually submitting it for execution by IRE.

Use identification simulation to construct an input payload, and simulate processing of the payload by IRE. You can then examine the results, adjust identification rules if needed, and re-run the simulation of the updated payload.

Use the identification simulation to:

- Automatically construct input payload that is based on existing identification rules, hosting and containment rules.
- Simulate execution of a payload (automatically constructed by identification simulation, or manually created).
- Browse payload output and execution log messages for a simulated run.

Note:

- Identification simulation does not commit any updates to the CMDB.
- Identification simulation supports simulation of processing payloads that are provided and which contain non-CMDB tables, but doesn't support the generation of such payloads.

Automatically generate payload using identification simulation

Use identification simulation to automatically construct an input payload for a specified class. The constructed payload is complete with any required dependent CIs, correctly structured, and syntactically valid for processing by the identification and reconciliation engine (IRE).

Before you begin

Role required: admin

About this task

The payload that is constructed during identification simulation is for the specified class. For a dependent CI class, you will be prompted for information about all dependencies. After you provide the required details, identification simulation constructs the payload based on your input.

Note: Automatically generating payloads that contain non-CMDB tables, isn't supported.

Procedure

1. Navigate to **All > Configuration > Identification/Reconciliation**, and click **Identification Simulation**.
2. In the **Start with CI Class** box click **Start**.
3. On the **Payload Information** form, in the **Data source** field, select the data source that is associated with this class update.
For the ServiceNow Discovery data source, select **ServiceNow**.

4. Select the **Class** in the payload.
 - a. In the **Criterion Attributes** area select the CI identifier attributes and then specify the values that uniquely identify a CI.
 - b. In the **Additional Attributes** area specify attributes and values that matching CIs will be updated with.
5. For dependent CIs associated with dependent identification rules, fill out the **Criterion Attributes** and **Additional Attributes** sections in all **Container level** sections that display.
6. Click **Generate Script**.
7. If any errors indicate that there are missing fields, fill in the missing fields and then click **Generate Script** again.

What to do next

- Click **Run Simulation** to simulate processing of the payload by IRE.
- Examine the results of the simulation, fine-tune the payload as needed, and combine with other payloads for other classes as desired. After finalizing the payload, use the `createOrUpdateCI()` API to execute the payload by IRE which will result in actual updates to the CMDB.
- Click **Copy Script** to copy the JSON script into the clipboard. You can then paste that script into a third party software or to another screen of the identification simulation.

Simulate payload processing using identification simulation

Use identification simulation to simulate the identification and reconciliation engine (IRE) process of CI identification for an input payload. Provide a valid payload, which was constructed using identification simulation or that was created manually.

Before you begin

Role required: admin

Procedure

1. Navigate to **All > Configuration > Identification/Reconciliation**, and click **Identification Simulation**.
2. (Optional) To run a simulation of an existing payload:
 - a. Click **Start** in the **Start with Existing Payload** tile.
 - b. On the Insert JSON Payload page, select the **Data source** that is associated with this class update.
 - c. (Optional) Select **Use Enhanced Identification** to apply the `identifyCIEnhanced` API for enhanced CI identification, instead of using the `identifyCI` API.
 - d. Paste the JSON payload into the empty canvas.
3. (Optional) To construct a new payload click **Start** in the **Start with CI Class** tile.
See [Automatically generate payload using identification simulation](#) for more information.
4. Click **Run Simulation** to simulate processing of the payload by IRE.

What to do next

1. Examine the results of the simulation in the results pane, and fine-tune the payload as needed:
 - a. Click **Run #1** to display the **Context ID** and the **Run ID** of the simulated run.
 - b. Click the drop down arrow next to **Run #1** to display additional details.
 - **Input:** Displays the payload for the simulation.
 - **Logs:** Displays all the logged messages that IRE generated while simulating processing of the payload, according to the specified logging level.
 - **Output:** Displays the output payload returned by IRE.

2. After finalizing the payload, use the [createOrUpdateCI\(\)](#) API to execute the payload by IRE which will result in actual updates to the CMDB.

Set logging level for identification simulation

Identification simulation logs each step of a simulated payload processing. You can then examine these run logs to determine if a payload was processed as expected, and if identification rules are effective. You can adjust the level of logging so it is helpful, and so that the amount of messages is not excessive or insufficient.

Before you begin

Role required: admin

Procedure

1. Navigate to **All > Configuration > Identification/Reconciliation**, and click **Identification Simulation**.
2. Click the **Settings** icon.
3. Select logging level for the identification and reconciliation engine (IRE) under **IE Log Level** and for the service cache under **Service Cache Log Level**.
The logging levels are displayed in ascending order, from the minimum level to the maximum level of logging.
4. Click on the **Settings** icon again to close the **Settings** dialog box.

Examine run logs

Identification simulation provides run logs which are generated by Identification and Reconciliation Engine (IRE). You can access these run logs for payload runs, to examine results and for debugging purposes. IRE payload output logs appear in a user friendly format on a central page.

Before you begin

Role required: admin

About this task

Also, internal applications that use IRE (such as Discovery) can call an internal API to provide a URL to viewing IRE run logs.

Logging is in the context of a specific run of the identification engine, and you can filter the log list by a specific data source and time range. Up to 1000 run logs that are up to 2 months old are listed, grouped by Context IDs, and run times. You can use the `glide.identification_logs.max_run_ids` property to modify the 1000 limit.

You can control the logging level by using the `glide.discovery.identification.log_level` Discovery system property and setting the value to one of the following:

- Info
- Warn
- Error
- Debug
- DebugVerbose
- DebugObnoxious

Note: IRE performs an initial verification of a payload before processing identification rules. If IRE detects any duplicate CIs based on any class identifiers, the payload is rejected and processing stops.

Procedure

1. Navigate to **All > Configuration > Identification/Reconciliation > Identification Logs**.
2. Filter the runs list as follows:
 - a. **Source:** Select the data source for which to display run logs.
 - b. **Time Range:** Specify a time range for which to display run logs. The **Runs** list displays all runs for the specified data source, during the specified time range.

3. In the **Runs** list, click a **Run #** to display its **Context ID** and **Run ID**.
A unique Context ID is associated with each specific payload that is run. Each run of that payload, is associated with a unique Run ID. A single Context ID for a payload that is run multiple times is associated with multiple Run IDs.
4. Click the drop down arrow for a **Run #** to display additional details.
 - **Input:** Displays the payload for the run.
 - **Logs:** Displays all the logged messages that the identification engine generated while running the payload, according to the specified logging level.
 - **Output:** Displays the output payload returned by the identification engine.

IRE error messages

The Identification and Reconciliation Engine (IRE) generates the following errors and messages. Depending on settings, these messages appear in the Identification Logging pane and in the system logs.

For information about lookup-based CI identification and qualifier chains, see [Create or edit a CI identification rule](#).

Note: IRE performs an initial verification of a payload before processing identification rules. If IRE detects any duplicate CIs based on any class identifiers, the payload is rejected and processing stops.

For information about CMDB Identification Payload error: "FAILED TRYING TO EXECUTE ON CONNECTION", see [CMDB Identification Payload error - "Insertion failed with error Error during insert of cmdb_ci..."](#), where node logs show "FAILED TRYING TO EXECUTE ON CONNECTION" "Duplicate entry 'XXX' for key 'XXX'" knowledge base article.

Error- IDENTIFICATION_RULE_MISSING

Message	Description and Resolution
Identity Rule Missing for table [xyz]	<p>Description: Identification rule is missing for a class.</p> <p>Resolution: Ensure that there is an identification rule for table [xyz], and that the rule is active.</p>

MISSING_MATCHING_ATTRIBUTES

Message	Description and Resolution
In payload missing minimum set of input values for criterion (matching) attributes from identify rule for table [xyz]. Add these input values in payload item 'abc'	<p>Description: Missing minimum set of values for criterion attributes for an identification rule.</p> <p>Resolution: In the payload, add minimum set of values for criterion attributes for CI Identifier for table [xyz]. Open the CI Class Manager, click Hierarchy and select the [xyz class]. Check the identification rule and the identifier entries for table [xyz].</p>

Error- NO_CLASS_NAME_FOR_INDEPENDENT_CI

Message	Description and Resolution
Cannot have 'sys_class_name' as a key field in an Independent Identity Rule on 'xyz'	<p>Description:</p> <p>The class attribute was added to the CI identifier which is not supported.</p> <p>Resolution:</p> <p>Remove the class attribute from CI Identifier for table [xyz].</p>

Error- IDENTIFICATION_RULE_FOR_LOOKUP_MISSING

Message	Description and Resolution
Identity Rule for table [xyz] missing Lookup Rule for class [abc]	<p>Description:</p> <p>The payload has a lookup class name, but the corresponding lookup rule is missing.</p> <p>Resolution:</p> <p>Add lookup identifier entry with [Search on table] as [abc] for CI Identifier for table [xyz].</p>

Error- IDENTIFICATION_RULE_FOR RELATED_ITEM_MISSING

Message	Description and Resolution
Identity Rule for table [xyz] missing Related Rule for class [abc]	<p>Description:</p> <p>The payload has a related class name, but the corresponding related rule is missing.</p> <p>Resolution:</p> <p>Add related entry with [Related table] as [abc] within CI Identifier for table [xyz].</p>

Error- NO_LOOKUP_RULES_FOR_DEPENDENT_CI

Message	Description and Resolution
Cannot have Lookup Rule for a Dependent Identity Rule on 'xyz'	<p>Description:</p> <p>Cannot have Lookup Rule for a Dependent Identity Rule.</p> <p>Resolution:</p> <p>Remove lookup identifier entry from dependent CI Identifier for table [xyz].</p>

Error- INVALID_INPUT_DATA

Message	Description and Resolution
<p>Found invalid sys_id in payload. No record with sys_id [xyz] exist in table [abc] or is a duplicate record with [duplicate_of] field set to a main CI</p>	<p>Description: The payload has a reference to an invalid sys_id.</p> <p>Resolution: Remove the referenced sys_id, or provide a valid sys_id.</p>
<p>In payload no data source exist. You need to provide choice value from choice field [discovery_source] in table [cmdb_ci]</p>	<p>Description: In payload no data source exists.</p> <p>Resolution: In the payload, provide a valid choice value from choice field [discovery_source] from table [cmdb_ci].</p>
<p>In payload invalid data source [xyz] exist. You need to provide a valid choice value from field [discovery_source] in table [cmdb_ci]</p>	<p>Description: The payload contains an invalid data source.</p> <p>Resolution: In the payload, provide a valid choice value from choice field [discovery_source] from table [cmdb_ci].</p>

Message	Description and Resolution
No such relationship with name [xyz] exist in table [cmdb_rel_type]. If out-of-box relationship for [xyz] has been removed or renamed, it should be restored	<p>Description:</p> <p>The payload is referencing a relationship that does not exist in the [cmdb_rel_type] table.</p> <p>Resolution:</p> <p>Verify that the reference to the relationship is accurate. Or, if it is a new relationship, add it to the [cmdb_rel_type] table. Or, If out-of-box relationship for [xyz] has been removed or renamed, restore it.</p>
Payload relations 'xyz' has invalid parent record index: [0]	<p>Description:</p> <p>Payload references invalid parent indexes.</p> <p>Resolution:</p> <p>Check payload indexes and ensure that they are all valid.</p>
Payload relations 'xyz' has invalid child record index: [0]	<p>Description:</p> <p>Payload references invalid child indexes.</p> <p>Resolution:</p> <p>Check payload indexes and ensure that they are all valid.</p>

Error- DUPLICATE_RELATIONSHIP_TYPES

Message	Description and Resolution
Duplicate relationship type records exists with name [xyz] in table [cmdb_rel_type] having sys_ids: [abc]	<p>Description: There are duplicate records in the [rel_ci_type] table for the relationship.</p> <p>Resolution: Remove the duplicate records.</p>

Error- DUPLICATE_PAYLOAD_RECORDS

Message	Description and Resolution
Found duplicate items in the payload (index 0 and 1), using className [xyz] and fields [abc]. Remove duplicate items from payload	<p>Description: The payload contains two items whose criterion attributes have identical values.</p> <p>Resolution: Remove one of the duplicate items.</p>

Error- LOCK_TIMEOUT

Message	Description and Resolution
Failed to acquire synchronization lock for xyz	<p>Description:</p> <p>Failed to acquire the system mutex lock.</p> <p>Resolution:</p> <p>Increase the mutex expiration time by adding the system property glide.identification_engine.mutex_expiration_time and setting to an integer value that is greater than the default value (15 min).</p>

Error- MULTIPLE_DUPLICATE_RECORDS

Message	Description and Resolution
Found duplicate records in table [xyz] using fields [abc]	<p>Description:</p> <p>Found duplicate records in the specified table.</p> <p>Resolution:</p> <p>Fix the duplicate records found by the identification engine. Check de-duplication tasks for information about all duplicates.</p>

Error- REQUIRED_ATTRIBUTE_EMPTY

Message	Description and Resolution
Missing mandatory field [xyz] in table [abc]. Add input value for mandatory field in payload	<p>Description:</p> <p>A required attribute is missing in the payload.</p> <p>Resolution:</p> <p>In the payload, add input value for mandatory field [xyz] in table [abc].</p>

Error- MISSING_DEPENDENCY

Message	Description and Resolution
In payload no relations defined for dependent class [xyz] that matches any containment/hosting rules: [abc]. Add appropriate relations in payload for 'def'	<p>Description:</p> <p>No relations defined for the dependent class that matches any of its metadata rules.</p> <p>Resolution:</p> <p>In payload add appropriate relations for dependent class [xyz] that matches any containment/hosting rules: [abc].</p>

Error- METADATA_RULE_MISSING

Message	Description and Resolution
No containment or hosting rules defined for dependent class [xyz]. Add containment/hosting rules for 'abc'	<p>Description:</p> <p>There are no containment or hosting rules defined for dependent class.</p> <p>Resolution:</p> <p>Add containment or hosting rules for dependent class [xyz].</p>

Error- MULTIPLE_DEPENDENCIES

Message	Description and Resolution
Found multiple dependent relation items [xyz] and [abc] in payload	<p>Description:</p> <p>Multiple dependent relation items exist.</p> <p>Resolution:</p> <p>Remove one of the multiple dependent relation items [xyz] or [abc].</p>
Multiple paths leading to the same destination: xyz -> abc	<p>Description:</p> <p>Multiple paths leading to the same destination.</p>

Message	Description and Resolution
	<p>Resolution:</p> <p>Remove duplicate relationship/qualifier chains that might exists between xyz -> abc.</p>

Error- ABANDONED

Message	Description and Resolution
Abandoning processing payload item 'xyz', since its depends on payload item 'abc' has errors	<p>Description:</p> <p>Dependent payload item has errors, so abandoning processing.</p> <p>Resolution:</p> <p>Resolve the error on the dependent payload item 'abc'.</p>
Can't find matched record with sys_id [xyz] in table [abc]	<p>Description:</p> <p>Matched sys_id does not exist in the corresponding table.</p> <p>Resolution:</p> <p>Check in table [abc] whether matched record is a valid record based on input payload.</p>
Identification engine API got called recursively, aborting...	<p>Description:</p> <p>The Identification engine API was called recursively.</p>

Message	Description and Resolution
	<p>Resolution:</p> <p>Avoid calling the Identification engine API recursively.</p>
Detected error while processing payload from xyz	<p>Description:</p> <p>Error occurred during processing payload.</p> <p>Resolution:</p> <p>Resolve all errors mentioned in the output payload from xyz.</p>
While processing relations encountered errors in payload item: xyz	<p>Description:</p> <p>Payload item has errors.</p> <p>Resolution:</p> <p>Resolve errors in payload item 'xyz'.</p>
Error occurred during parsing input json payload: xyz	<p>Description:</p> <p>Error occurred during parsing JSON payload.</p> <p>Resolution:</p> <p>Ensure that input JSON payload has correct JSON format.</p>

Error- MULTI_MATCH

Message	Description and Resolution
<p>Duplicate dependent records found having relationship [xyz] with same CI (className:[abc], sysId: [def])</p>	<p>Description: Found duplicate dependent CIs.</p> <p>Resolution: Check de-duplication tasks for information about all duplicates, and then delete duplicate records.</p>
<p>Found multiple relations between payload items: 'xyz' and 'abc'</p>	<p>Description: Found multiple relations between payload items.</p> <p>Resolution: Check for duplicate relationship chains and qualifier chains that might exist.</p>
<p>Found duplicate records in lookup table [xyz] using fields [abc] and reference field [def]</p>	<p>Description: Found duplicate records in lookup table.</p> <p>Resolution: Check de-duplication tasks for information about all duplicates, and then delete duplicate records.</p>

Error- QUALIFICATION_LOOP

Message	Description and Resolution
Qualification chain has loop that contains relation 'xyz'	<p>Description: Qualification chain has a loop.</p> <p>Resolution: Remove the loop from the qualification chain with relation 'xyz'.</p>

Error- TYPE_CONFLICT_IN_QUALIFICATION

Message	Description and Resolution
Invalid payload, qualification chain has multiple possible paths for payload items: 'xyz' and 'abc'	<p>Description: Multiple qualification paths found.</p> <p>Resolution: Remove multiple possible qualification paths between items 'xyz' and 'abc'.</p>

Error- RECLASSIFICATION_NOT_ALLOWED

Message	Description and Resolution
CI Reclassification not allowed from class: [xyz] to [abc]	<p>Description: CI reclassification not allowed.</p> <p>Resolution: Check reclassification tasks for information about reclassification, and check if reclassification from class: [xyz] to [abc] is valid.</p>

Error- DUPLICATE_RELATED_PAYLOAD

Message	Description and Resolution
Found duplicate Related items (0 and 1) in the payload index 1 using fields xyz	<p>Description: Duplicate Related items present.</p> <p>Resolution: Remove one of the duplicate related items present in the payload.</p>

Error- DUPLICATE_LOOKUP_PAYLOAD

Message	Description and Resolution
Found duplicate Lookup items (0 and 1) in the payload index 1 using fields xyz	<p>Description: Duplicate lookup items present.</p> <p>Resolution: Remove one of the duplicate lookup items present in the payload.</p>

INSERT_NOT_ALLOWED_FOR_SOURCE

Message	Description and Resolution
Insert into [xyz] is blocked for data source [abc] by IRE data source rule	<p>Description: An IRE data source rule is configured to prevent data source [abc] from inserting CIs of the [xyz] class.</p> <p>Resolution: Delete or update the appropriate IRE data source rule to let data source [abc] insert CIs of the [xyz] class. Or, wait for another permitted data source to create the same CI.</p>

CI reclassification during IRE processing

During the Identification and Reconciliation Engine (IRE) CI identification process, a CI might need to be reclassified to a different sys_class_name type. By default, CIs are reclassified automatically. If automatic reclassification is disabled, then the CI is not reclassified and the system generates a reclassification task for your review.

The class of a CI can be upgraded, downgraded, or switched to a different branch in the class hierarchy. For more details about reclassification operations, see [Reclassify a CI](#). You can use system properties and payload flags to configure the IRE behavior of CI reclassification, globally or individually per CI.

Note: CI reclassification is possible only between two classes that have identical identification rules.

Configure automatic CI reclassification using system properties

You can use system properties to configure system-wide IRE behavior for CI reclassification. For information about CI reclassification-related properties, including access, see [Properties for Identification and Reconciliation](#).

-

The following properties enable or disable automatic reclassification updates that are specified in a payload. These properties are set to **true** in the base system, enabling processing of CI updates, including CI reclassification updates.

To disable any automatic reclassification update, set the respective property to **false**. In that case, IRE rejects a payload (or a payload item in Enhanced IRE) with the respective reclassification updates, and creates a [recognition task](#).

- glide.class.upgrade.enabled
- glide.class.downgrade.enabled
- glide.class.switch.enabled

-

The following properties enable IRE to process CI updates with reclassification operations. However, depending on the property setting, IRE processes or skips the reclassification update. These properties are set to **false** in the base system, in which case IRE processes CI updates including any CI reclassifications.

Set a property to **true** to configure IRE to process CI updates but not the CI respective reclassification update.

- `glide.identification_engine.update_without_switch_enabled`
 - `glide.identification_engine.update_without_downgrade_enabled`
 - `glide.identification_engine.update_without_upgrade_enabled`
- This set of properties takes precedence over the previous set of properties (`glide.class.<reclassification>.enabled`). For example, with the following conflicting property settings, the second property takes precedence over the first:
- `glide.class.downgrade.enabled = false`
 - `glide.identification_engine.update_without_downgrade_enabled = true`

Example for IRE processing of a payload item with a switch of a CI from Linux Server to Window Server. With the following default property settings in the base system, IRE updates the attributes including the class switch:

- `glide.class.switch.enabled = true`
- `glide.identification_engine.update_without_switch_enabled = false`

However, with the following property settings, IRE updates the attributes but skips the class switch:

- `glide.class.switch.enabled = true`
- `glide.identification_engine.update_without_switch_enabled = true`

Configure automatic CI reclassification in input payloads

You can use flags which correspond to the system properties, in the input payload of the `CreateOrUpdateCIEnhanced()` or the `createOrUpdateCI()` APIs. In the payload, set these flags to **true** or **false**

to temporarily override the respective system property settings, at the payload item level.

Also, you can pass payload level settings (which apply to all items within a payload), per data source, by specifying CI reclassification properties on the Robust Import Set Transformers form. For more information, see [Robust import set transformer properties](#).

Payload flags that control reclassification behavior:

- `classUpgrade`
- `classDowngrade`
- `classSwitch`
- `updateWithoutUpgrade`
- `updateWithoutDowngrade`
- `updateWithoutSwitch`

The following sample JSON payload enables automatic reclassification for the specified CI:

```
{ "items": [{className: 'cmdb_ci_server', classUpgrade: true, classDowngrade: true, classSwitch: true, values: {name: 'linux123', serial_number: '12srt567', ip_address: '10.2.3.4'}, }]}
```

Reclassification restriction rules

Prevent IRE from downgrading or switching a CI class during payload processing to help prevent data loss. A reclassification restriction rule prevents a CI class change for specific source and target classes, while still processing any other property updates for the CI.

You can use a reclassification restriction rule, for example, to prevent a CI class downgrade from `cmdb_ci_linux_server` (source class) to `cmdb_ci_server` (target class). Or, to prevent a CI class switch from Linux Server to Windows Server. Reclassification restriction rules can be useful when using a Service Graph Connector which might lead to a class downgrade or switch, and a potential loss of important data.

To control the application of reclassification restriction rules:

- Use the `glide.identification_engine.reclassification_restriction_rules_enabled` system property to globally enable or disable the application of active reclassification restriction rules. This property is set to **true** by default.
- Use the `skipReclassificationRestrictionRules` payload flag in an IRE payload to prevent the application of active reclassification restriction rules.

For example, a payload with the `skipReclassificationRestrictionRules` flag:

```
{  
  "items": [  
    {  
      "className": "cmdb_ci_server",  
      "values": {  
        "short_description": "Linux server description",  
        "name": "Linux Server 1"  
      },  
      "settings": {  
        "skipReclassificationRestrictionRules": "true"  
      }  
    }  
  ]  
}
```

For information about how to create a reclassification restriction rule, see [Create a reclassification restriction rule](#).

Create a reclassification restriction rule

Reduce data loss during IRE processing by preventing a CI class change for specific source and target classes. A reclassification restriction rule affects only the Class attribute and does not prevent the update to the rest of the CI properties.

Before you begin

Role required: Itil_admin (Itil has read privilege only)

About this task

If during IRE processing of a payload, a CI needs to be reclassified (downgrade or switch class), IRE checks reclassification restriction rules. If any reclassification restriction rule applies to the current CI reclassification, IRE processes the CI properties update, but skips the CI reclassification.

IRE output provides specific details about any processing related to reclassification restriction rules.

A reclassification restriction rule applies only to the direction between the specified source and the target classes. The rule doesn't prevent a reclassification in the opposite direction, from the specified target class to the source class. To restrict reclassification between two classes in both directions, specify two separate reclassification restriction rules, one for each direction.

Procedure

1. Enter `cmdb_ire_reclassification_restriction.list` in the filter navigator.
2. Fill out the Reclassification Restriction form.

Field	Description
Name	Name of the reclassification restriction rule.
Source table	Current CI class.
Source inheritance	Whether to apply the reclassification restriction rule to child classes of Source table .
Target class	Reclassification target class.
Target inheritance	Whether to apply the reclassification restriction rule to child classes of Target table .

Field	Description
Type	CI reclassification type: Downgrade or Switch .

3. Click **Submit**.

What to do next

In the Reclassification Restrictions list view, you can activate or deactivate a reclassification restriction rule by setting its **Active** value to true or false.

View a reclassification task

When automatic CI reclassification is disabled, reclassification tasks are created for CIs that could not be automatically reclassified during the identification process. Review these tasks to locate the CIs and decide if to reclassify them.

Before you begin

Role required: admin or itil

Procedure

1. Navigate to **All > Configuration > Identification/Reconciliation > Reclassification Tasks**.
2. Select a reclassification task.
3. Examine the details on the Reclassification Task form.

Reclassification Task form

Field	Description
Configuration item	The CI that must be reclassified.
Short description	Short description noting that CI reclassification was not allowed.

Field	Description
Description	Description noting the current class of the CI and the class that the CI must be changed to.
Internal payload	Payload used in the identification process.

What to do next

After examining the task details, you can locate the CI that is noted in the task **Description** and manually reclassify it. For details, see [Reclassify a CI](#).

CMDB dependent relationship rules

Service definitions consist of CI types and relationship types. Dependent relationship rules define the dependency structure of the CI types and the relationship types in these service definitions, helping in CI identification and in the construction of business service maps.

The dependencies that are defined by these rules are used when identifying dependent CIs to prioritize the order of CI identification, and to match CIs and respective dependent CIs in a payload. Dependent relationship rules are also used by Service Mapping and can be defined for custom CI types. After defining a new CI type, you can define dependent relationship rules that specify how the new CI type is related to existing types in the CMDB.

Dependent relationship rules consist of hosting and containment rules (dependent relationship rules), each type modeling the data from a different perspective of the CI. Containment rules represent CIs' configuration hierarchy, describing which CI contains which other CIs. Hosting rules represent CIs' placement in a business definition, describing what CIs run on.

Both hosting and containment rules describe a relationship type between two CI types and the same relationship type can be used in a hosting rule and in a containment rule. It is the context in which the relationship is used that distinguishes between a containment and hosting rule.

Manage dependent relationship rules:

- To access rules at the class level, use the CI Class Manager. Navigate to **All > Configuration > CI Class Manager**.
- To access grouped rules, use the Metadata Editor. Navigate to **All > Configuration > Identification/Reconciliation > Metadata Editor**.

The plugins that have been activated on an instance determine which hosting and containment rules exist in a base system.

Hosting rules

Hosting rules represent all the possible valid combinations of pairs of hosting and hosted CIs in the service definition. Hosting rules are a flat set of rules that can be only one level deep, and which always involve resources, typically physical or virtual hardware. Each hosting rule is a stand-alone rule between two CI types, describing either a valid CI type that another CI type can host, or by which another CI type can be hosted. A hosting rule consists of a parent CI type, a relationship type (such as Hosted On::Hosts) and a child CI type. For example, you can have a hosting rule that specifies that the CI type 'Application' 'Runs On::Runs', the CI type 'Hardware'.

A CI can be hosted on multiple resources (such as Windows and Linux). This CI is represented by a hosting rule for the CI with each resource that the CI can be hosted on. During CI identification, the pair of CIs that are being examined, should satisfy at least one hosting rule.

Hosting rules are stored in the CMDB Metadata Hosting Rules [cmdb_metadata_hosting] table.

Containment rules

Containment rules represent the containment hierarchy for a CI type, describing valid objects that a CI type can contain in the service definition, and valid objects that can be contained by the CI type. Containment rules are chained to each other in a containment rules group, with a CI type that is the top-level (root) parent of the group. The collection of containment rules construct a hierarchy-like map of containment relationships. Containment rules are logical concepts used to represent logical CIs, for example to describe software that runs on a server. A containment rule consists of a parent CI type, a relationship type (such as 'Contained By::Contains'), and a child CI type.

For example, you might have a containment rule specifying that the CI type 'Tomcat' 'Contains::Contained By' CI type 'WAR File'.

Endpoints are special containment rules that specify incoming or outgoing connections in the model, designating the CI types that data of some specified type flows in to or out from the service definition. After adding an endpoint to a containment rule, you cannot add any child rules to the endpoint rule.

Containment rules are stored in the CMDB Metadata Containment Rules [cmdb_metadata_containment] table.

Reference rules

Reference rules are used mostly by Cloud Management to represent all of the possible valid combinations of pairs of referencing and referenced CIs in the service definition.

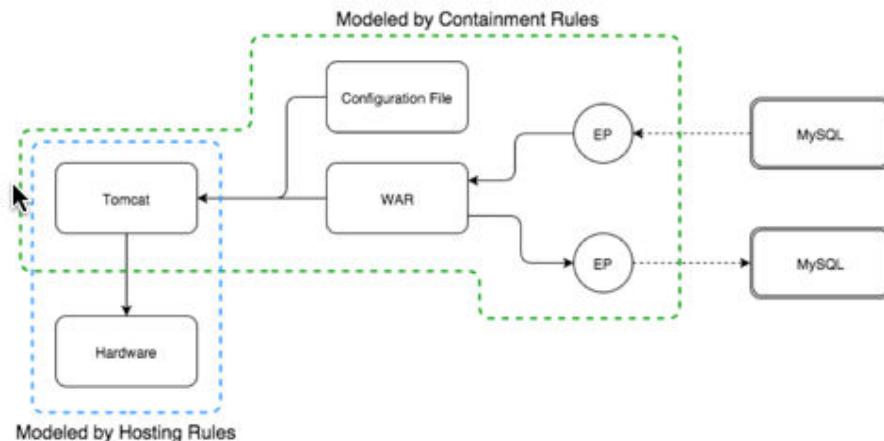
- Reference rules are a flat set of rules that can be only one level deep.
- Reference rules always involve resources, typically virtual entities. Each reference rule is a stand-alone rule between two CI types, describing either a valid CI type that another CI type can reference, or by which another CI type can be referenced. Both the CI classes should be able to live independent of each other.
- A referencing rule consists of a parent CI type, a relationship type (such as `Provisioned From::Provisioned`) and a child CI type. For example, you can have a referencing rule that specifies that the CI type 'Virtual Machine' `Provisioned From::Provisioned`, the CI type 'Image'.
- A CI can reference multiple resources (for example, a VM Instance can have a reference relation with both the Image and the Hardware templates). This CI is represented by a referencing rule for the CI with each resource that the CI can be referenced from.
- The reference rule cannot be part of the CI identification.
- Reference rules are stored in the CMDB Metadata Reference Rules [cmdb_metadata_reference] table.

Rules requirements

The rules that you create are bound by the following requirements which narrow the relationships and ensure that only valid options are available in the drop-down lists in the Metadata Editor.

- Given a CI type that is as a child in a containment rule: Not this CI type or its children can be a top-level (root) parent of any other containment rule, and it cannot be in any hosting rule, either as a parent or as a child.
- Given a CI type that is a top-level (root) parent of a containment rule: It cannot be a child in a hosting rule (for example, you cannot be hosted on Tomcat, if Tomcat has any containment rules).
- Given a CI type that is a child in a hosting rule: It cannot be in any containment rule, either as a parent or a child.
- Given a CI type that is a parent in a hosting rule: It cannot be a child in any containment rule.
- Hosting rules cannot create loops such as Tomcat –runs_on- VMWare –runs_on- Tomcat.

Example: Hosting and containment rules model



Hosting rules that model the diagram:

Tomcat 'Runs on' Hardware.

Containment rules that model the diagram:

- Tomcat 'Contains' Configuration File
- Tomcat 'Contains' WAR
- WAR has two endpoints for JDBC with MySQL:
 - Inbound
 - Outbound

Example: Valid set of rules

```
Tomcat Hosted Linux  
Linux Hosted Computer
```

The second metadata entry triggers the third requirement, which is satisfied (it is a hosting rule, not a containment rule).

- [Create dependent relationship rules](#)

Create hosting and containment rules (dependent relationship rules) for CI classes to help with correctly identifying dependent CIs during the business discovery process and service mapping. Discovery calls the identification API that applies dependent relationship rules.

Create hosting and containment rules (dependent relationship rules) for CI classes to help with correctly identifying dependent CIs during the business discovery process and service mapping. Discovery calls the identification API that applies dependent relationship rules.

You can create a basic hosting or containment rule in the CI Class Manager. Or, use the Metadata Editor to create groups of hosting and containment rules, and inbound or outbound endpoints in containment rules. The CI Class Manager and the Metadata Editor are synchronized, and you can use each of those tools to display and edit a dependent rule.

Create a dependent relationship rule for a CMDB class

Use the CI Class Manager to create a basic dependent relationship rule (hosting or containment relationship rule) for a CMDB class.

Before you begin

Role required: itil has read access, itil_admin (on top of itil) has full access.

About this task

The class for which you create dependent relationship rule, must have a [dependent identification rule](#).

Procedure

1. Navigate to **All > Configuration > CI Class Manager**.
2. Click **Hierarchy** to display the CI Classes list, and select the class for which you want to create a hosting or a containment rule.
3. In the class navigation bar, click **Dependent Relationships**.
4. In the Dependent Relationships view, click **Add dependency**.
5. Fill out the details in the Add Dependent Relationship Rule dialog box.

Field	Description
Rule Type	Designation of whether this rule is a hosting rule or a containment rule.
This Class	The class that the rule applies to.
Relationship	The relationship type for the rule.
Target Class	The target class for the dependent relationship rule. The designation of this class as a child or parent class, is based on the specified Relationship .

6. Click **Save**.

What to do next

You can click **Reset to derived** and then confirm the operation to delete all dependent relationship rules that were added specifically for the selected class. Only dependent relationships that are derived from a parent class, remain.

For more information about child and parent classes, see [Table extension and classes](#).

Create or edit a collection of containment rules

Create containment rule for CIs to help with correctly identifying dependent CIs during the business discovery process and service mapping. Discovery calls the identification API that applies dependent relationship rules.

Before you begin

Role required: admin

About this task

A containment rule is a dependent relationship rule which defines a relationship between two CIs, structured as: CIType1 RelationshipType CIType2. The first CI type that you add becomes the top level CI of a containment rules group which is a chain of containment rules. The entire set of containment rules is organized as groups according to top-level CIs.

To create a containment rules group for a new CI type, you need to first add the CI Type1 of the relationship. To add a child containment rule for a CI type that exists, you need to select that CI type, and define the second portion of the relationship rule which is the relationship type and CI Type2.

To each rule within a containment rules group you can add inbound or outbound endpoints, which are noted by blue up and down arrows. After adding an endpoint, you can not add a containment rule in that branch of the containment rules hierarchy.

Procedure

1. Navigate to **All > Configuration > Metadata Editor**.
2. In the Metadata Editor, click the **Containment Rules** tab.
3. Click **Add New Rule** to add a top-level rule or point to a rule for which you want to add a child rule and click the green '+' icon that appears on the right.
4. Complete the **Add Containment Rule to <class>** form.

Field	Description
Configuration Item Type	The CI class that the rule applies to.
Relationship Type	The relationship type for the rule.
Reverse Relationship Direction	Enable to use the reverse relationship in the rule.
Always include in Service Model	Enable to always include the CIs of the specified class in the Service Map if their parent CI (based on the containment relationship) is present in the Service Map.

5. Click **Create**.
6. Add an endpoint to a child rule:
 - a. Point to a child rule for which you want to add an endpoint.
 - b. Click the blue "+" icon that appears on the right.
 - c. Complete the **Add Endpoint To <class>** form.

Field	Description
Endpoint Type	The type of endpoint.

Field	Description
Inbound or Outbound	The direction of the endpoint.

- d. Click **Create**.

Create or edit a collection of hosting rules

Create hosting rule for CIs to assist in correctly identifying dependent CIs during the business discovery process and service mapping. Discovery calls the identification API that applies dependent relationship rules.

Before you begin

A hosting rule is a dependent relationship rule which defines a relationship between two CIs, structured as: <CI Type1> <relationship type> <CI Type2>. To create a hosting rule, you need to add a CI type as <CI Type1> in the relationship rule, and then define the second portion of the relationship rule which is the relationship type and <CI Type2>. The entire set of hosting rules is organized as groups according to the top-level hosted CIs.

A hosting rule implicitly contains two rules, which are the reversal of each other. When you create the rule '<CI Type1> <relationship type> <CI Type2>', the rule '<CI Type2> <reversed relationship type> <CI Type1>' is automatically added.

Role required: admin

Procedure

1. Navigate to **All > Configuration > Metadata Editor**.
2. In the Metadata Editor, click the **Hosting Rules** tab.
3. Click **Add New Rule** to add a top-level rule or point to a rule for which you want to add a child rule and click the green '+' icon that appears on the right.
4. Complete the **Add Hosted/Hosting Rule to <class>** form.

Field	Description
Configuration Item Type	The <CI Type2> in the rule.
Relationship Type	The relationship type for the rule.
Reverse Relationship Direction	Check to reverse relationship in the rule.

5. Click **Create**.

IRE support for non-CMDB tables

Apply Identification and Reconciliation Engine (IRE) processes to supported non-CMDB tables to ensure data integrity and health of those tables.

Starting with the Vancouver release, IRE supports some non-CMDB tables. You can use all IRE features with some non-CMDB tables after creating identification rules (CI identifiers and identifier entries) for those tables. Non-CMDB tables supported for IRE features include:

- In an application-specific scope: All non-CMDB tables
- In the global scope: Only non-CMDB tables that are preset in the base system. In the Vancouver release for example, the Location [cmn_location], Department [cmn_department], Cost Center [cmn_cost_center], Building [cmn_building], User [sys_user], and Group [sys_user_group] non-CMDB tables are supported.

You can't use the [CI Class Manager](#) to manage any IRE-related rules for non-CMDB tables. Instead, you must work directly with the respective tables in list views to create and manage those rules as described in the following procedures:

- [Create an identification rule for a non-CMDB table](#)
- [Create a reconciliation rule for a non-CMDB table](#)
- [Create an IRE data source rule for non-CMDB tables](#)
- [Create a data refresh rule for a non-CMDB table](#)

- Create an identification inclusion rule for a non-CMDB table
- Simulate payload execution using identification simulation
- Use partial payloads
- Review and remediate de-duplication tasks

You can use the following store apps with supported non-CMDB tables:

- CMDB 360 in CMDB Workspace
- IntegrationHub ETL

IRE processes are applied to supported non-CMDB tables with the following differences:

- IRE doesn't populate the discovery_source, last_discovered, and the first_discovered attributes if those attributes don't exist in the non-CMDB table.
- IRE uses the non-CMDB table's class name as sys_class_name if the table doesn't include a sys_class_name attribute.
- IRE payloads don't support relationships with non-CMDB tables.

Note: Although IRE-related user interface and accompanying documentation might reference CMDB and CMDB elements, most of those references also apply to any supported non-CMDB tables.

- Create an identification rule for a non-CMDB table

To use Identification and Reconciliation Engine (IRE) features with supported non-CMDB tables, you must first create identification rules that uniquely identify the table records. Each non-CMDB table can be associated with a single identification rule.

- Create a reconciliation rule for a non-CMDB table

Create a static or a dynamic CI reconciliation rule for a non-CMDB table.

- Create a data refresh rule for a non-CMDB table

To apply Identification and Reconciliation Engine (IRE) features to supported non-CMDB tables, create data refresh rules for those tables. Data refresh rules are used to determine if a record is stale for a specific data source. Such records can then be updated by a lower-priority authorized data source.

- [Create an identification inclusion rule for a non-CMDB table](#)

Narrow the scope of records that are included in the identification process of non-CMDB records by creating an identification inclusion rule.

- [Create an IRE data source rule for non-CMDB tables](#)

When using Identification and Reconciliation Engine (IRE), you can prevent a specific data source from inserting new records for a specific non-CMDB table. Create IRE data source rules for data sources that you don't trust in creating records but continue to trust in updating those records that exist.

To use Identification and Reconciliation Engine (IRE) features with supported non-CMDB tables, you must first create identification rules that uniquely identify the table records. Each non-CMDB table can be associated with a single identification rule.

Before you begin

Role required: itil has read access, itil_admin (on top of itil) has full access

About this task

Each identification rule consists of a single identifier for the table, one or more identifier entries, and one or more related entries.

Review the following topics before creating identification rules:

- [Identification rules](#)
- [Effective usage of CMDB Identification](#)

When creating identifier entries, you can configure the **Search on table** and **Criterion attributes** fields on the Identifier Entry form to implement one of the following options:

Regular identifier entry

Lets you select attributes from the associated identifier table.

Lookup identifier entry

Lets you select attributes from any related table (Lookup table), other than the currently selected table.

Hybrid identifier entry

Lets you select attributes from both the currently main selected table, and from another table (Lookup table).

For non-CMDB tables, only independent identification rules are supported.

Procedure

1. Navigate to **All > Identification/Reconciliation > CI Identifiers**.
2. In the Identifiers list view, click **New**.
3. Fill out the Identifier form.

Field	Description
Name	Name of CI identifier.
Applies to	Supported non-CMDB table.
Independent	Must be checked to indicate that the identifier can identify a record independently of other records.

4. Click **Submit**.
5. In the Identifiers list view, locate and open the identifier that you just created.
6. On the Identifier form, select the **Identifier Entries** tab and then click **New**.

7. Fill out the Identifier Entry form.

Field	Description
Identifier	Preset with the name of the table of the associated identifier.
Search on table	<p>Preset with the label of the table of the associated identifier.</p> <p>To create:</p> <ul style="list-style-type: none">• A regular identifier entry: Set to the identifier table and select Criterion attributes from that same table.• A lookup identifier entry: Set to another table (lookup table) and select Criterion attributes from that lookup table.• A Hybrid identifier entry: Set to another table (lookup table) and then do the following steps.<ul style="list-style-type: none">• Select Criterion attributes from the lookup table.• Add Hybrid Entry CI Criterion Attributes from the current table using background scripts, after saving the rule. For more details, see the 'What to do next' section at the end of this task.

Field	Description
	A lookup table should have a reference to the associated identifier table.
Criterion attributes	<p>Set of attributes that uniquely identify the record. Attributes can belong to the current class, or to a parent class.</p> <p>Note: It's possible to add reference fields as a criterion attribute. However, such fields might not always be effective:</p> <ul style="list-style-type: none"> • Reference fields store sys_ids that point to a record in another table, and thus is considered a weak criterion attribute (in terms of uniqueness) for the current table. • The system detects and then replaces invalid values in a reference field with 'Unknown'. For example, an invalid Model ID value is replaced with the value 'Unknown'. Also, if several CIs end up having that same reference field set to 'Unknown', then these CIs become duplicate CIs.
Priority	Priority of applying the identifier entry. Rules with lower priority numbers are given higher

Field	Description
	<p>priority. Identifier entries of identical priorities are applied randomly.</p> <p>You can keep gaps between the priority numbers, so you can assign the unused priority numbers to new entries without modifying the existing priority order.</p>
Active	<p>Specifies whether the identifier entry is active. At least one identifier entry in an identification rule must be active for the rule to apply.</p>
Enforce exact count match	<p>For lookup identification, match a record only on exact lookup records count match.</p> <p>When enforced, all lookup items for a record in the payload must have matching records in the lookup table that reference the same record:</p> <ul style="list-style-type: none"> a. Only matches records that have all the lookup items from the input payload referencing the record in the table. b. If there are multiple matches, selects the oldest created record as the final match. <p>When not enforced, one lookup item for a record in the payload matching a record in the lookup table, is sufficient to consider a match:</p>

Field	Description
	<ul style="list-style-type: none"> a. Matches any record that has at least one of the lookup items from the input payload referencing the record in the table. b. If there are multiple matches, selects the records with the max number of lookup items from the input payload referencing the record in the table. c. If there are still multiple matches, selects the oldest created record as the final match.
Allow null attribute	<p>When selected, then if at least one criterion attribute isn't null, attempt matching with an identifier entry even if there are criterion attributes that are null.</p> <p>Otherwise, all criterion attributes must have values to attempt matching with an identifier entry.</p>
Allow fallback to parent's rules	Allows the identification rules of the record's parent table to be used if a match isn't found for this identification rule. Applies only for dependent identification rules.
Optional condition	A filter to narrow the set of records that will be searched for a matching record.

Field	Description
	<p>Available only if the <code>glide.identification_engine.enabled_identifier_optional_condition</code> system property is set to true (false by default). In the base system, identifier entries of various classes are pre-configured with advanced options conditions. All these pre-configured conditions in regular identifier entries will automatically apply when you set this property to true. Therefore, to prevent unexpected behavior, review those predefined conditions in regular identifier entries before setting this property to true.</p> <p>For more details about this property, see Properties for Identification and Reconciliation.</p>

Note: If criterion attributes have only two attributes and `sys_class_name` is one of them (for example `[name, sys_class_name]`, `[ip_address, sys_class_name]`), then the other attribute can't be NULL, even if **Allow null attribute** is enabled. This restriction is due to `sys_class_name` being considered a special system matching attribute.

8. Click **Submit**.
9. On the Identifier form, select the **Related Entries** tab and then click **New**.
10. Fill out the Related Entry form.

Related Entry form

Field	Description
Identifier	Preset with the identifier that this related entry is associated with.
Active	Check box that specifies that the related entry is active.
Related table	A related table (lookup table) that references the record that is being matched.
Referenced field	A referenced field in Related table with a reference to the associated identifier table.
Criterion attributes	The set of attributes to uniquely identify the related item. Attributes can belong to the current class, or to a parent class.

Field	Description
	<p>Note: It's possible to add reference fields as a criterion attribute. However, such fields might not always be effective:</p> <ul style="list-style-type: none"> Reference fields store sys_ids that point to a record in another table, and thus is considered a weak criterion attribute (in terms of uniqueness) for the current table. The system detects and then replaces invalid values in a reference field with 'Unknown'. For example, an invalid Model ID value is replaced with the value 'Unknown'. Also, if several CIs end up having that same reference field set to 'Unknown', then these CIs become duplicate CIs. <p>Click the lock icon to view, add, or remove attributes from the identification rule.</p>
Allow null attribute	If at least one criterion attribute in the related table isn't null, allow to attempt matching with an identifier entry even if there are criterion attributes which are null.

Field	Description
Priority	<p>Priority of the related entry for the specified Related table. Rules with lower priority numbers are given higher priority while matching a related item for a specific related table. Related entries for the specified related table with identical priorities are applied randomly.</p> <p>You can keep gaps between the priority numbers, so you can assign the unused priority numbers to new entries without modifying the existing priority order.</p>
Optional condition	Filter conditions to narrow the set of records that will be searched for a matching related item.

11. Click **Submit**.

What to do next

To add criterion attributes to a **Hybrid Entry CI Criterion Attributes** field in a hybrid identifier entry, instead of using the Identifier Entry form, you must use background scripts. After saving the identification rule, navigate to **System Definitions > Scripts - Background**, and then enter a script that adds the attributes and click **Run script**.

Sample script:

```
var gr = new GlideRecord('cmdb_identifier_entry');
// get the identifier entry you want to update
gr.get('<identifier_entry_sys_id>');
// set the attributes you want in the hybrid rule in a comma separated list
// for example: 'name,serial_number'
gr. hybrid_entry_ci_criterion_attributes='<column_name_1>
```

```
,<column_name_2>,<etc.>' ;  
gr.update();
```

This process requires the admin role.

Create a static or a dynamic CI reconciliation rule for a non-CMDB table.

For information about static reconciliation rules, dynamic reconciliation rules, and other principals related to reconciliation rules, see [Reconciliation rules](#).

If both, static and dynamic reconciliation rules exist for the same record attribute, the dynamic rule has precedence.

Note: You can't create a reconciliation rule for system fields or for Identification and Reconciliation Engine (IRE) specific fields such as the Discovery source (discovery_source) field. Also, reconciliation rules can't be dot-walked using reference fields.

Create a static reconciliation rule for a non-CMDB table

A static reconciliation rule specifies class attributes that data sources are authorized to update, and prevents unauthorized data sources from overwriting the attributes' values. A static reconciliation rule also specifies the prioritization among multiple data sources. Without static reconciliation rules, data sources can overwrite each other's updates to attribute values.

Before you begin

Role required: itil has read access, itil_admin (on top of itil) has full access.

About this task

Static reconciliation rules are used in conjunction with [data refresh rules](#) to determine reconciliation steps for a record. These rules determine if, when, and by which data source a record can be updated. If multiple data sources are authorized to update the same attributes, assign a priority to each of these data sources to prevent them from overwriting each other's updates.

After an authorized data source updates an attribute, subsequent updates are accepted only from the same data source or from a data

source with a higher priority. Updates from a data source with a lower priority are rejected, unless these two conditions are met:

- The lower priority source is the first source updating the record.
- The record became stale based on data refresh rules for the class.

Precedence order of static reconciliation rules:

- Rule configured for a specific attribute, has precedence over rule set with **Apply to all attributes** (regardless of priority value).
- Between two rules for the same attribute or between two rules set with **Apply to all attributes**, the rule that is specific directly for the class has precedence over the derived rule.
- Between two rules for the same attribute or between two rules set with **Apply to all attributes** at the same class level, precedence is determined by rule priorities.

Information about the last discovery source that updates each attribute is stored in the Data Source History [cmdb_datasource_last_update] table, but only after enabling the reconciliation rule. Therefore, there might be unexpected updates after you enable the rule until the highest priority data source has updated the CI.

Static reconciliation rules affect reconciliation of stale attributes. During reconciliation, the information in the Data Source History table is considered along with the data refresh rules for the CI's class, to determine if a CI attribute is stale. A CI attribute is determined to be stale if it was not updated by the latest discovery source to update the CI, within a time period. The time period is specified by the Effective Duration time in the data refresh rule for the class for the discovery source. In this case, if another authorized discovery source, with a lower priority attempts to update the stale CI attribute, the update is allowed.

If there is a dynamic reconciliation rule for the same record attribute as in a static reconciliation rule, the dynamic rule takes precedence.

Procedure

1. Navigate to **All > Configuration > Identification/Reconciliation > Reconciliation Definitions**.

2. In the Reconciliation Definitions list view, click **New**.
3. Fill out the Reconciliation Definition form.

Field	Description
Data source	The data source that you are configuring this rule for.
Priority	Priority of Data source within other data sources for the specified attributes. Smaller numbers designate higher priority. Data sources without a reconciliation rule are assigned the lowest priority.
Applies to	Authorizes the specified data source to update all attributes of the specified non-CMDB table. Note: This setting will be overridden by any setting that applies to a specific attribute. In which case, instead of using this option, you can directly include all attributes for Attributes .
Filter condition	Conditions that records must meet for the rule to be applicable. For example, to apply this rule only to records that are associated with the Finance department, select this condition: [Department] [is] [Finance] .

Field	Description
	<p>Note: The <code>glide.identification_engine.enable_reconciliation_filter_before_update</code> system property determines when filter conditions are applied. By default, those filter conditions are applied after attribute values have changed during payload processing. Set this property to true so that Identification and Reconciliation Engine (IRE) applies the filter conditions before attribute values change.</p>
Attributes	<p>Attributes from the current or from a parent class, that the specified data source is authorized to update.</p> <p>Available only if Apply to all attributes is not selected.</p>
Update with null	<p>Attributes that the specified data source can update with a null value. By default, authorized data sources cannot overwrite a non-null value with a null value.</p> <p>Attributes in this list, which are not in the Attributes list, are not included with the attributes that the data source can update with a null value.</p>

4. Click **Submit**.

Create a dynamic reconciliation rule for a non-CMDB table

A dynamic reconciliation rule for non-CMDB table uses CMDB 360 data to choose a value such as the largest value that is reported, for updating a record.

Before you begin

CMDB 360/Multisource CMDB must be enabled.

Role required: itil has read access, itil_admin (on top of itil) has full access

About this task

If the same CI attribute has both, a static reconciliation rule and a dynamic reconciliation rule, the dynamic reconciliation rule has precedence.

A dynamic reconciliation rule supports several rule types, such as largest reported value and most reported value. When applying a dynamic reconciliation rule, IRE processes the current payload and then examines the CMDB 360 data store to select a value with which to update the CMDB. Depending on the dynamic reconciliation rule type, selecting the appropriate value might not be immediately conclusive. For example, there might not be a single value that is most reported, or for some values, the last discovered timestamp isn't reported. Therefore, when necessary, IRE falls back to examining additional details such as last reported, last discovered, and last updated values to select the most appropriate value.

Note: You can't add a dynamic reconciliation rule when creating a new child class in the CI Class Manager. You must first save the new child class and then add the dynamic reconciliation rule.

Procedure

1. Click **All**.

2. In the Filter navigator, enter `cmdb_dynamic_reconciliation_definition.list` to open the Dynamic Reconciliation Definitions table.
3. In the Dynamic Reconciliation Definitions list view, click **New**.
4. Fill out the Dynamic Reconciliation Definition form.

Field	Description
Name	
Attributes	Attributes for which to apply the rule. Attributes that the specified rule type can't be applied to and attributes for which a dynamic reconciliation rule already exists for, don't appear.
Filter condition	Conditions that CIs must meet for the rule to be applicable. For example, to apply a rule only to CIs that are associated with the Finance department, select this condition: [Department] [is] [Finance] .
Applies to	Non-CMDB table that this rule applies to.
Dynamic Rule Type	Rule type which is based on CMDB 360 data.

5. Click **Submit**.

To apply Identification and Reconciliation Engine (IRE) features to supported non-CMDB tables, create data refresh rules for those tables. Data refresh rules are used to determine if a record is stale for a specific

data source. Such records can then be updated by a lower-priority authorized data source.

Before you begin

Role required: itil has read access, itil_admin (on top of itil) has full access

About this task

Data refresh rules have no impact when dynamic reconciliation rules are in effect.

Data refresh rules are used in conjunction with static reconciliation rules to determine reconciliation steps for a record. These rules determine if, when, and by which data source a record can be updated.

Procedure

1. Click **All**.
2. In the Filter navigator, enter `cmdb_datasource_staleness.list` to open the Data Source Staleness Definitions table.
3. In the Data Source Staleness Definitions list view, click **New**.
4. Fill out the Data Source Staleness Definitions form.

Field	Description
Applies to	Non-CMDB class that this rule applies to.
Data source	Data source for which record staleness is evaluated.
Effective Duration	The time period that is used for the staleness evaluation. If the fields specified in the static reconciliation rule for the record's class were not updated by the specified data source within the specified time period

Field	Description
	— the record is determined to be stale for that data source. If you enter a value with a prefix that is valid and a suffix that is not, such as 15 x — the valid portion of the value is used ('15'). If the entire value is invalid — the default value of 0 is used.
Active	Activates the rule.

5. Click **Submit**.

Narrow the scope of records that are included in the identification process of non-CMDB records by creating an identification inclusion rule.

Before you begin

Role required: itil has read access, itil_admin (on top of itil) has full access.

About this task

During duplication detection of independent Cls, the Identification and Reconciliation Engine (IRE) processes only the records that satisfy the identification inclusion rules. For example, you can set a filter to include only records whose state is operational. When no identification inclusion rules exist, all records are included in the identification process.

In the base system, there are no predefined identification inclusion rules. Identification inclusion rules are defined at the class level.

Note: Identification inclusion rules impact any script that calls IRE, therefore create them carefully. Identification inclusion rules can prevent the identification of certain types of records, affecting some features of Discovery and Service Mapping.

Procedure

1. Navigate to **All > Configuration > Identification/Reconciliation > Identification Inclusion Rules**.
2. In the Identification Inclusion Rules list view, click **New**.
3. Fill out the Identification Inclusion Rules form.

Field	Description
Applies to	Non-CMDB table that this rule applies to.
Inclusion condition	Criteria that non-CMDB records must meet to be included in the identification process.

4. Click **Save**.

When using Identification and Reconciliation Engine (IRE), you can prevent a specific data source from inserting new records for a specific non-CMDB table. Create IRE data source rules for data sources that you don't trust in creating records but continue to trust in updating those records that exist.

Before you begin

Role required: itil_admin

About this task

IRE data source rules have no impact when dynamic reconciliation rules are in effect.

- Child classes derive IRE data source rules from parent classes like identification rules do.
- IRE data source rules that are specified for a child class, override any IRE data source rules derived from a parent class.

When IRE processes an insert operation that is prohibited by an IRE data source rule, the insert operation fails. This failure happens when

the data source and record class in the insert operation and in an IRE data source rule, match. When `CreateOrUpdateCIEnhanced()` is used, IRE stores the failed payload in the CMDB IRE Partial Payloads [cmdb_ire_partial_payloads] table for future potential use.

Note: When an insert operation is not allowed by the IRE data source rule, then when using `createOrUpdateCI()`, the entire IRE payload fails since `createOrUpdateCI()` doesn't allow partial commits.

If later, a permitted data source attempts to insert that same record, then IRE inserts the record after merging it with the matching record from the partial payloads. IRE then deletes the partial payload from the CMDB IRE Partial Payloads [cmdb_ire_partial_payloads] table, and allows future updates by the data source specified in the rule.

IRE data source rules do not apply to lookup and related items, and only a single rule can be active for any class/data source pair.

Procedure

1. Navigate to **All > Configuration > Identification/Reconciliation > IRE Data Source Rules**.
2. In the IRE Data Source Rules list view, click **New** and fill out the IRE Data Source Rule form.

Field	Description
Data source	Data source that is not allowed to create CIs of the specified class.
Active	Activates the IRE data source rule.
Applies to	The class (and child classes) that the specified data source is not allowed to create records for.
Insert Not Allowed	Disables the specified data source from inserting new

Field	Description
	records of the specified class, to the non-CMDB table.

3. Click **Submit**.

Result

If a payload item with an insert request, and in which the data source and the record class match the data source and the record class specified in the IRE data source rule:

1. The insert operation fails and IRE logs the following message:
INSERT_NOT_ALLOWED_FOR_SOURCE Insert into [xyz] is blocked for data source [xyz] by IRE data source rule.
2. If using [CreateOrUpdateCIEnhanced\(\)](#), then IRE stores the payload item as a partial payload in the CMDB IRE Partial Payloads [cmdb_ire_partial_payloads] table.

If later, a permitted data source successfully inserts a record that matches the record from a partial payload item:

1. The current record is merged with the matching record from the partial payload, applying static reconciliation rules as needed.
2. The respective partial payload in the CMDB IRE Partial Payloads [cmdb_ire_partial_payloads] table is deleted.
3. Later payloads in which the non-permitted data source updates the respective record, run successfully.
4. IRE allows the data source, that was previously prohibited from inserting the record, to update that same record which now exists in the non-CMDB table.

Effective usage of CMDB Identification

Use CMDB Identification effectively.

Identification rules

An independent identification rule identifies a CI based on the CI's attributes, independently of other CIs.

A dependent identification rule identifies a CI by its dependent CIs and the relationships of the identified CI with those dependent CIs. Identification with a dependent identification rule is based on the dependent CIs and the relationships and qualifiers between the identified CI and its dependent CIs. Identification then requires more time than with an independent identification rule and is prone to some identification errors. Usage of dependent rules should therefore be minimized.

CI modeling determines which type of identification rules are required for proper CI identification.

Create identification rules using the following order of importance:

1. Independent identification rules — It is always preferable to create independent identification rules rather than dependent identification rules. When you model a CI, define the CI with a complete set of attributes that lend themselves to independent identification, eliminating the need to use additional CIs for identification.
2. Dependent identification rules — If it is necessary to create dependent identification rules, then define a single level of dependency. Two is the maximum number of dependency levels that is supported.
3. Avoid creating lookup identifier entries. The use of lookup identifier entry is highly discouraged as it can reduce performance. If unavoidable, ensure to first review class definitions and consider updates that allow usage of independent identification rules.
4. Limit the number of identifier entries within an identification rule, ideally to 1. A second identifier entry can further reduce performance, as will each additional identifier entry.
5. Create strong identification rules in which the strongest identifier entries and related entries are set with the highest priority.
6. Ensure that the identification rule is at the class level that it needs to be.

Payload

Create the payload using the following order of importance:

1. Payload size — Limit the number of Cls per payload to 500.

2. Avoid duplicate entries in the payload.

Example: If an identification rule has a criterion attribute for the **name** field, then the following payload has duplicate items resulting in failure:

```
var payload = {
    items: [
        {
            className: 'cmdb_ci_linux_server',
            values: {
                name: 'Win Server 200',
                ram: '2048'
            }
        },
        {
            className: 'cmdb_ci_linux_server',
            values: {
                name: 'Win Server 200',
                ram: '4096'
            }
        }
];
```

3. Do not pass system data such as the following in the payload.

```
var payload = {
    items: [
        {
            className: 'cmdb_ci_linux_server',
            values: {
                name: 'Win Server 200',
                sys_domain: 'global',
                sys_domain_path: 'xyz',
                sys_updated_on: '2017-06-15 16:25:11',
                sys_mod_count: 23,
            }
        }
];
```

4. Provide the minimum necessary set of criterion attributes for each payload item, according to what is specified in the corresponding identification rules.

- When matching CIs, use CIs' sysIds if available. If provided, IRE can use the sysId to directly locate a CI without requiring any criterion attributes from the identification rule. In this case, IRE does not use the sysId in the matching process.

- Example: Independent CI that needs to be updated — sysId is available.

```
var payload = {  
    items: [{  
        className: 'cmdb_ci_linux_server',  
        values: {  
            sys_id: '194876usytrr65378098',  
            ram: '2048',  
        }  
    }]  
};
```

- Example: Dependent CI that needs to be inserted. Tomcat War CI depends on Tomcat CI, and Tomcat CI depends on Linux Server CI. Syslibs for the Tomcat and the Linux CIs are available.

```
var payload = {
    items: [
        {
            className: 'cmdb_ci_app_server_tomcat_web',
            values: {
                name: 'war1',
                short_description: 'my description'
            }
        },
        {
            className: 'cmdb_ci_app_server_tomcat',
            values: {
                sys_id: '194876usytrr65378098'
            }
        },
        {
            className: 'cmdb_ci_linux_server',
            values: {
                sys_id: '09876tysueyt6345lakiu'
            }
        }
    ],
    relations: [
        {
            parent: 1,
            child: 0,
            type: 'Contains::Contained by'
        }
    ]
}
```

```
, {  
    parent:1,  
    child:2,  
    type:'Runs on::Runs'  
}; ]
```

- Example: Dependent CI that needs to be updated — sysId is available.

```
var payload = {  
    items: [{  
        className:'cmdb_ci_app_server_tomcat_wa  
r',  
        values: {  
            sys_id:'039387euey637465sytet',  
            short_description:'my description n  
ew'  
        }  
    }]  
};
```

6. When inserting many CIs, all of which depend on the same CI, you should serialize your API calls. Otherwise, attempting to concurrently process many CIs can clog the system, significantly degrading overall system performance.

Properties for Identification and Reconciliation

Use the Identification and Reconciliation properties to configure the identification and reconciliation engine (IRE).

These properties are available for Identification and Reconciliation. To view and edit these properties, the admin role is required.

Note: To open the System Properties [sys_properties] table, enter sys_properties.list in the navigation filter.

Properties for Identification and Reconciliation

Property	Description
Enforce the requirement that required attributes cannot be	<ul style="list-style-type: none">Type: true false

Property	Description
null during identification and reconciliation. glide.required.attribute.enabled	<ul style="list-style-type: none"> Default value: true Location: Configuration > CMDB Properties > Identification/Reconciliation Properties
Allow class upgrade during IRE identification and reconciliation. glide.class.upgrade.enabled	<ul style="list-style-type: none"> Type: true false Default value: true Location: Configuration > CMDB Properties > Identification/Reconciliation Properties Learn more: CI reclassification during IRE processing. <p>When false, IRE rejects a payload (or a payload item in Enhanced IRE) with the respective reclassification update, and creates a recategorization task.</p>
Allow class downgrades during IRE identification and reconciliation. glide.class.downgrade.enabled	<ul style="list-style-type: none"> Type: true false Default value: true Location: Configuration > CMDB Properties > Identification/Reconciliation Properties Learn more: CI reclassification during IRE processing. <p>When false, IRE rejects a payload (or a payload item in Enhanced IRE) with the</p>

Property	Description
	respective reclassification update, and creates a reclassification task .
Allow class switching during IRE identification and reconciliation. <code>glide.class.switch.enabled</code>	<ul style="list-style-type: none"> Type: true false Default value: true Location: Configuration > CMDB Properties > Identification/Reconciliation Properties Learn more: CI reclassification during IRE processing. <p>When false, IRE rejects a payload (or a payload item in Enhanced IRE) with the respective reclassification update, and creates a reclassification task.</p>
<code>glide.identification_engine.update_without_upgrade_enabled</code>	Enable IRE to process CI updates with upgrade reclassification updates. This property takes precedence over the <code>glide.class.upgrade.enabled</code> property. <ul style="list-style-type: none"> Type: true false Default value: false Location: Add to System Properties [sys_properties] table. Learn more: CI reclassification during IRE processing.

Property	Description
	<p>Depending on the property setting, IRE processes or skips the upgrade update:</p> <ul style="list-style-type: none"> • true: IRE processes the CI updates but doesn't process the CI upgrade reclassification update. • false: IRE processes the CI updates including the CI upgrade reclassification update.
glide.identification_engine.update_without_downgrade_enabled	<p>Enable IRE to process CI updates with downgrade reclassification updates. This property takes precedence over the glide.class.downgrade.enabled property.</p> <ul style="list-style-type: none"> • Type: true false • Default value: false • Location: Add to System Properties [sys_properties] table. • Learn more: CI reclassification during IRE processing.
	<p>Depending on the property setting, IRE processes or skips the downgrade update:</p> <ul style="list-style-type: none"> • true: IRE processes the CI updates, but doesn't process the CI downgrade reclassification update. • false: IRE processes the CI updates including the

Property	Description
	CI downgrade reclassification update.
glide.identification_engine.update_without_switch_enabled	<p>Enable IRE to process CI updates with switch reclassification updates. This property takes precedence over the glide.class.switch.enabled property.</p> <ul style="list-style-type: none"> Type: true false Default value: false Location: Add to System Properties [sys_properties] table. Learn more: CI reclassification during IRE processing. <p>Depending on the property setting, IRE processes or skips the switch update:</p> <ul style="list-style-type: none"> true: IRE processes the CI updates, but doesn't process the CI switch reclassification update. false: IRE processes the CI updates including the CI switch reclassification update.
glide.identification_engine.reclassification_restriction_rules_enabled	Globally enable or disable the application of active reclassification restriction rules. <ul style="list-style-type: none"> Type: true false Default value: true

Property	Description
	<ul style="list-style-type: none"> Location: Add to System Properties [sys_properties] table. Learn more: CI reclassification during IRE processing.
Allow the update of an empty field by a lower priority data source. <code>glide.reconciliation.override.null</code>	<ul style="list-style-type: none"> Type: true false Default value: true Location: Configuration > CMDB Properties > Identification/Reconciliation Properties
Controls how identification processes a small set of duplicate CIs. <code>glide.identification_engine.skip_duplicates</code>	<ul style="list-style-type: none"> Type: true false Default value: true Other values: true <p>If the number of duplicate CIs is less than the threshold specified by <code>glide.identification_engine.skip_duplicates.threshold</code>, then the oldest of the duplicate CIs is picked as a match and gets updated. That oldest CI is also designated as the main CI for that set of duplicate CIs.</p> <p>For the rest of the duplicate CIs, the <code>duplicate_of</code> field is set as a reference to the main CI.</p>

Property	Description
	<p>false</p> <p>Matching a CI fails, and an error is logged.</p> <ul style="list-style-type: none"> Location: Configuration > CMDB Properties > Identification/Reconciliation Properties
<p>Maximum number of CIs that can be in a set of duplicate CIs to allow identification to process the duplicate CIs according to the setting of <code>glide.identification_engine.skip_duplicates</code>.</p> <p><code>glide.identification_engine.skip_duplicates.threshold</code></p>	<p>If the number of duplicate CIs exceeds the threshold, then identification processes the duplicate CIs as if <code>glide.identification_engine.skip_duplicates</code> is set to false.</p> <ul style="list-style-type: none"> Type: Integer Default value: 5 Location: Configuration > CMDB Properties > Identification/Reconciliation Properties
<p>Maximum number of log runs that can be displayed when navigating to Configuration > Identification Logs.</p> <p><code>glide.identification_logs.max_run_ids</code></p>	<ul style="list-style-type: none"> Type: integer Default value: 1000 Location: Configuration > CMDB Properties > Identification/Reconciliation Properties
<p><code>glide.cache.size.service_cache</code></p>	<p>Maximum cache size (in MB) that is used by the identification engine for inbound and outbound relations. When the limit is reached, the least recently</p>

Property	Description
	<p>used cached data is discarded, releasing space for new data.</p> <p>Note: You cannot disable the service cache.</p> <ul style="list-style-type: none"> • Type: Integer • Default value: 20 • Location: Add to System Properties [sys_properties] table.
glide.identification_engine.granular_insert_locking	<p>Determines whether to use multiple granular insert locks or single global insert lock.</p> <p>Set to false if there are performance issues associated with the usage of multiple granular insert locks.</p> <ul style="list-style-type: none"> • Type: true false • Default value: true • Location: Add to System Properties [sys_properties] table.
glide.identification_engine.batch_update_last_discovered	<p>Controls batch update of last_discovered field in CIs that are being processed by the identification engine.</p> <p>Set to false if there are business rules that apply to the last_discovered field, and you want to trigger these rules</p>

Property	Description
	<p>when calling an Identification and Reconciliation API.</p> <ul style="list-style-type: none"> • Type: true false • Default value: true • Location: Add to System Properties [sys_properties] table.
glide.identification_engine.related_items_local_cache_count	<p>For optimization, a custom number of locally cached query result entries of related/lookup items.</p> <ul style="list-style-type: none"> • Type: integer • Default value: 15000 • Location: Add to System Properties [sys_properties] table. <p>Note: If there is a memory issue due to optimization related to using local cache, set the glide.identification_engine.related_items_local_cache_count and the glide.identification_engine.dependent_items_local_cache_count properties to 0.</p>
glide.identification_engine.dependent_items_local_cache_count	<p>For optimization, a custom number of locally cached query result entries of dependent Cls.</p> <ul style="list-style-type: none"> • Type: integer • Default value: 10000

Property	Description
	<ul style="list-style-type: none"> Location: Add to System Properties [sys_properties] table. <p>Note: If there is a memory issue due to optimization related to using local cache, set the <code>glide.identification_engine.related_items_local_cache_count</code> and the <code>glide.identification_engine.independent_items_local_cache_count</code> properties to 0.</p>
<code>glide.identification_engine.independent_items_local_cache_count</code>	<p>For optimization, a custom number of locally cached query result entries of independent Cls.</p> <ul style="list-style-type: none"> Type: integer Default value: 100000 Location: Add to System Properties [sys_properties] table. <p>Setting the value to 0 avoids using local cache for independent Cls which might affect performance.</p>
<code>glide.cmdb.logger.source.identification_engine</code>	<p>Enable and configure what type of details the system logs when using IRE outside the scope of identification simulation. For example, when using an API, ECC queue or scheduled jobs.</p> <ul style="list-style-type: none"> Type: string

Property	Description
	<ul style="list-style-type: none">Values: info, warn, error, debug, or debugVerboseLocation: Add to System Properties [sys_properties] table. <p>Note: Depending on the setting, the system can generate large amounts of data that might affect overall system performance. Set the value with caution, and limit the level of details and use time to the minimum necessary for testing or debugging.</p> <p>For more troubleshooting information, see the How to capture IRE [identification and reconciliation engine] debug logs [KB0750382] knowledge base article.</p>
glide.identification_engine.partial_payload_items_max_size	<p>Maximum number of items allowed when creating a partial payload. When that limit is reached, the partial payload is split.</p> <p>For example, when IRE creates a partial payload, items and associated relations and references, are all merged in one partial payload. This merge could result in a large partial payload.</p>

Property	Description
glide.identification_engine.partial_items_process_limit	<p>Adjusting this property can help with performance issues related to IRE processing of partial items.</p> <ul style="list-style-type: none">• Type: integer• Default: 1000• Learn more: Identification and Reconciliation engine (IRE)• Location: Add to System Properties [sys_properties] table
glide.identification_engine.partial_items_process_limit	<p>Maximum number of partial items to be fetched in a single IRE call. After reaching this limit, IRE fetches only partial items corresponding to complete items in the input payload.</p> <p>Adjusting the value can help with performance issues related to IRE processing of partial items.</p> <ul style="list-style-type: none">• Type: integer• Default: 2000• Location: Add to System Properties [sys_properties] table.
glide.identification_engine.partial_items_process_absolute_limit	<p>Absolute limit of the number of partial items for IRE to fetch, after which, IRE stops fetching partial payloads from the CMDB IRE Partial Payloads [cmdb_ire_partial_payloads] table. Adjusting the value can help with</p>

Property	Description
	<p>performance issues related to IRE processing of partial items.</p> <ul style="list-style-type: none"> • Type: integer • Default: 5000 • Location: Add to System Properties [sys_properties] table.
<code>glide.identification_engine.skip_updating_source_last_discovered_if_older</code>	<p>Determines how IRE updates the last_discovered and the discovery_source attributes in the CMDB.</p> <ul style="list-style-type: none"> • true: If last_discovered is provided in the payload and it is older than the last_discovered of the CI in the CMDB, IRE does not use the payload values to update the last_discovered and the discovery_source attributes in the CMDB. • false: Even if the last_discovered provided in the payload is older than the last_discovered of the CI in the CMDB, IRE uses the payload values to update the last_discovered and the discovery_source attributes in the CMDB. <p>Note: Only the attributes mentioned above are affected by this property in an update operation.</p> <ul style="list-style-type: none"> • Type: true false

Property	Description
	<ul style="list-style-type: none"> • Default: true • Learn more: Identification and Reconciliation engine (IRE) • Location: Add to System Properties [sys_properties] table.
glide.identification_engine.ire_message_listener_skip_updating_source_last_discovered_to_now	<p>If Robust Transform Engine (RTE) does not pass the ire.skip_updating_last_scan_to_now custom property on the Robust Import Set Transformer form, IRE uses the value of this property for the skip_updating_source_last_discovered_to_now IRE option.</p> <ul style="list-style-type: none"> • Type: true false • Default: false • Location: Add to System Properties [sys_properties] table.
glide.identification_engine.skip_updating_last_scan_if_older	<p>Determines how IRE uses the source_recency_timestamp value in a payload to update the last_scan attribute in the Source [sys_object_source] table.</p> <ul style="list-style-type: none"> • true: If source_recency_timestamp is provided in the payload and it is older than the last_scan of the CI in the CMDB, IRE does not update the last_scan attribute in the Source [sys_object_source] table.

Property	Description
	<ul style="list-style-type: none"> • false: Even if the source_recency_timestamp provided in the payload is older than the last_scan of the CI in the CMDB, IRE uses the payload value to update the last_scan attribute in the Source [sys_object_source] table. <p>Note: Only the attributes mentioned above are affected by this property in an update operation.</p> <ul style="list-style-type: none"> • Type: true false • Default: true • Learn more: About mapping data columns to CMDB classes and attributes • Location: Add to System Properties [sys_properties] table.
glide.identification_engine.ire_message_listener_skip_updating_last_scan_to_now	<p>If RTE does not pass the ire.skip_updating_last_scan_to_now custom property on the Robust Import Set Transformer form, IRE uses the value of this property for the ire.skip_updating_last_scan_to_now IRE option.</p> <ul style="list-style-type: none"> • Type: true false • Default: false

Property	Description
	<ul style="list-style-type: none"> Learn more: About mapping data columns to CMDB classes and attributes Location: Add to System Properties [sys_properties] table.
glide.identification_engine.platform_domain_separation_enabled	<p>Toggles domain separation support mode during IRE processing.</p> <ul style="list-style-type: none"> false: IRE processes run only within the current domain. Basically disabling parent domains access to child domains during IRE processing. true: IRE domain separation follows the platform domain separation behavior. Basically, enabling parent domains to look access into all its child domains during IRE processing. <ul style="list-style-type: none"> Type: true false Default: false Learn more: Domain separation and CMDB Identification and Reconciliation Location: Add to System Properties [sys_properties] table.
glide.identification_engine.enable_identifier_optional_condition	<p>Enables advanced options for regular identifier entries in identification rules. Those advanced options let you add conditions to narrow the set of</p>

Property	Description
	<p>records that will be searched for a matching CI.</p> <p>Note:</p> <p>This property affects only regular identifier entries (it doesn't affect lookup or hybrid identifier entries).</p> <p>In the base system, identifier entries of various classes are pre-configured with advanced options conditions. All these pre-configured conditions in regular identifier entries will automatically apply when you set this property to true.</p> <p>To prevent unexpected behavior, review those predefined conditions in regular identifier entries before setting this property to true. In the Filter box in the primary navigation, enter <code>cmdb_identifier_entry.list</code>. Then, in the Identifier Entry list view, review the 'Optional condition' column.</p> <ul style="list-style-type: none">• Type: true false• Default: false• Learn more: Create or edit a CI identification rule

Property	Description
glide.identification_engine.enable_reconciliation_filter_before_update	<ul style="list-style-type: none">• Location: Add to System Properties [sys_properties] table.• Determines whether filter conditions of a reconciliation rule are applied before a value change during payload processing, or after.• Type: true false• Default: false• Learn more: Create a static reconciliation rule, Create a dynamic reconciliation rule• Location: Add to System Properties [sys_properties] table.

Components installed with Identification and Reconciliation

Several types of components are installed with Identification and Reconciliation (included in the com.snc.cmdb plugin), including tables.

Tables installed

Table	Description
Identifier [cmdb_identifier]	Identification rule sets defined for different classes of CIs.

Table	Description
Reconciliation Definition [cmdb_reconciliation_definition]	Static reconciliation rules defined for different classes of CIs at the table and field level.
Dynamic Reconciliation Definitions [cmdb_dynamic_reconciliation_definition]	Dynamic reconciliation rules defined for different attributes and classes.
Identifier Entry [cmdb_identifier_entry]	Rule entries with different priorities assigned to each identifier.
Duplicate Audit Result [duplicate_audit_result]	Duplicate audit results corresponding to a specific duplicate task. These results are generated automatically during the identification process and should not be added manually.
Remediate Duplicate Task [reconcile_duplicate_task]	Task to address duplication that is detected during the identification process. Records are generated automatically, and users should not add records manually.
Reclassification Task [reclassification_task]	Reclassification tasks that were generated during the identification process.
Data Source History [cmdb_datasource_last_update]	Information about the last data source that updated each attribute. Used to determine if a data source can update a stale CI.

Table	Description
Data Source Staleness Definition [cmdb_datasource_staleness]	Effective duration per data source. When effective duration is exceeded, then CMDB Health determines that the information provided by that data source is stale.
Identification Engine Context [cmdb_ie_context]	<p>Input payload, and data source (cmdb_ci's discovery_source) that will be used as input for a specific identification engine API. Stores information about which specific identification engine API will be called (identifyCI or createOrUpdateCI API). Also stores information about enhanced IRE options used in Identification Simulation.</p> <p>Note: Internal table used by identification simulation.</p>
Identification Engine Run [cmdb_ie_run]	<p>Specific cmdb_ie_context record that was used to run against the identification engine. Also details about the output payload returned by APIs, such as start and end time of the run and whether the run was successful.</p> <p>Note: Internal table used by identification simulation.</p>
Identification Engine Log	Identification engine logs for a specific cmdb_ie_run simulated in

Table	Description
[cmdb_ie_log]	the identification simulation. Also details about logs level and order. Note: Internal table used by identification simulation.
IRE Data Source Rule [cmdb_ire_data_source_rule]	IRE data source rules.
CMDB IRE Partial Payloads [cmdb_ire_partial_payloads]	Payload items that were determined to be partial, and which might be later matched with an incoming payload. If a partial payload is matched and processed, it is deleted from the CMDB IRE Partial Payloads table. Partial payloads older than 90 days are deleted from the table. For more information about usage of this table in IRE processes, see Identification and Reconciliation engine (IRE) .
CMDB IRE Partial Payloads Index [cmdb_ire_partial_payloads_index]	Identifier keys associated with partial items. IRE uses those keys to try to match with identifier keys of incoming payloads. For more information about usage of this table in IRE processes, see Identification and Reconciliation engine (IRE) .

Table	Description
<p>CMDB IRE Incomplete Payloads [cmdb_ire_incomplete_payloads]</p>	<p>Incomplete items, stored using JSON format as incomplete payloads. Incomplete items are stored for the purpose of logging payloads with irrecoverable errors, and are never processed again.</p> <p>The table is configured for table rotation, with duration of one day and seven table rotations.</p> <p>For more information about usage of this table in IRE processes, see Identification and Reconciliation engine (IRE).</p>
<p>IRE Output Aggregate Stats [cmdb_ire_output_aggregate_stats]</p>	<p>This table is populated when RTE invokes IRE, for example, when processing integrations.</p> <p>Details about data inserted by Import Sets or Robust Transform Engine (RTE) to the CMDB (via IRE). Numbers of items inserted, partial items, and updated items, are stored for each type of CI, per run.</p>
<p>IRE Output Target Items [cmdb_ire_output_target_item]</p>	<p>This table is populated when RTE invokes IRE, for example, when processing integrations.</p> <p>Details about data inserted by Import Sets or Robust Transform Engine (RTE) to the CMDB (via IRE). Target class and the sys_id are</p>

Table	Description
	stored per ImportSet row id, within a run. For this table to populate, RTE must pass the ire_output_detailed_stats property.
Reclassification Restrictions [cmdb_ire_reclassification_restriction]	Reclassification restriction rules. These rules prevent switch and downgrade reclassification updates for specific source and target classes. For more information, see CI reclassification during IRE processing .

User roles installed

Role	Description
cmdb_payload_admin	Automatically assigned to users with the cmdb_admin role for internal use only.

Related reference

- [Properties for Identification and Reconciliation](#)

Populating the CMDB

You can populate the CMDB by using Discovery, by using the IntegrationHub ETL or Import Sets to import and integrate data from a third-party source, by integrating with an external CMDB, or by manually creating CIs.

When you populate the CMDB with information, you create a record for each configuration item in the cmdb_ci table or on one of the tables that extend that table.

Related ServiceNow® Store apps and reference information:

- **CMDB schema model:** A collection of class diagrams and class attributes for key CMDB classes.
- **CMDB tables descriptions:** Descriptions of key CMDB tables in the base system.
- **CMDB CI Class Models:** A ServiceNow Store app that adds class models that extend the base CMDB class hierarchy. This includes class descriptions, identification rules, identifier entries, and dependent relationships if applicable. You can then use the added classes as any other CMDB base class.
- **Discovery patterns:** A ServiceNow Store app that provides a library of Discovery patterns for discovering specific devices and applications in the industry.
- **Service Graph Connectors:** ServiceNow Store apps that provide pre-defined integrations for importing and integrating common third-party data into CMDB classes. Also includes the [IntegrationHub ETL](#) wizard for creating new ETL transform maps.

ITIL configuration management auto-discovery

The key to any configuration management business practice is the initial and on-going inventory or discovery of what you own. The ServiceNow platform provides three options for auto-discovery:

- The separate and highly robust [Discovery](#) product.
- For organizations that want to leverage the discovery technologies they already have deployed (SMS, Tally NetCensus, LanDesk, and so on), the ServiceNow platform supports integrations to those technologies via web services. Scanned data can be mapped directly into the CMDB.

For further information on designing, constructing, and maintaining the CMDB, see the [CMDB Design & Configuration](#) white paper.

Discovery

The Discovery product automatically populates the CMDB. Discovery searches the network for all attached computers and devices, then populates the CMDB with information on each computer/device's configuration, provisioning, and current status. Discovery uses probes,

sensors, and patterns, to collect and process data about computers, servers, printers, a variety of IP-enabled devices, and the relationships between all the items found. Discovery also reports on any software which is running, and the TCP connections between computer systems, thereby establishing their relationships. This information is sent back to the instance and is used to populate the CMDB.

For more information about Discovery see:

- [ITOM Visibility](#)
- [Discovery basics](#)

Integrate third-party data using IntegrationHub ETL

Use the [IntegrationHub ETL](#) to import and integrate data from a third party into the CMDB. Using IntegrationHub ETL, create ETL transform maps which are used for integrating data from specific data sources. IntegrationHub ETL guides you through importing source data, transforming any data if needed, and selecting target CMDB classes and attributes to map the data to. You then preview the integration results and adjust any configurations before scheduling recurring integrations.

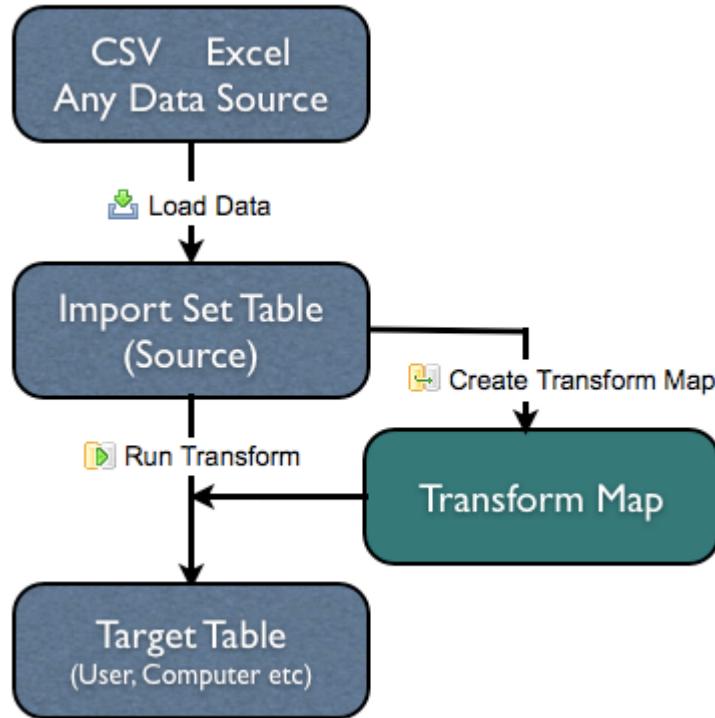
Visit the [ServiceNow Store](#) website to view and download common integrations.

Import data from another source using Import Sets

You can import data to the CMDB using Import Sets. [Import sets](#) find files of information (in formats such as XML, Excel, or CSV), import them, and transform them onto the required table. This process can be scheduled or performed on demand.

To import relationships between CIs, use import sets to populate the table [cmdb_rel_ci] with information on the parent, the child, and the nature of the relationship. The [cmdb_rel_ci] table displays a list of all CI relationships and is useful when importing CI data.

Import Sets overview



CMDB instance API

Use the [CMDB instance API](#) to populate the CMDB by creating or updating CMDB tables.

Manually create a CI

Create a single CI for a specific class. The role required is based on the settings of the class table you select for the CI.

1. Use the CI Class Manager:
 - a. Navigate to **All > Configuration > CI Class Manager**.
 - b. Click **Hierarchy** to display the list of CI Classes. Select the class to use for the CI.
 - c. In the class navigation bar, select **CI List** and then on the CI list view, click **New**.

- d. Fill out the CI form and then click **Submit**.
2. Or, directly use a table:
 - a. Navigate to **All > Configuration** and then elect the class to use for the CI, such as Business Services.
 - b. In the navigation filter of the application navigator, enter the table label (such as 'Linux'), or the table name in the format of <table name>.list (such as 'cmdb_ci_linux_server.list'). Then, press Enter.
 - c. In the list view of the table, click **New** and fill out the form fields for the table.
 - d. Click **Submit**.

Integrating third-party data into the CMDB

Import and integrate third-party data into CMDB classes and properties.

Available methods

Use the combination of the following data integration methods to integrate third-party data into CMDB.

Integration Commons for CMDB



Framework that provides a set of common operations and functionalities for integrations.

IntegrationHub ETL



ETL transform maps for integrations.

Service Graph Connectors



Connectors for bringing in third-party data correctly and quickly.

Related applications and features

Learn about applications and features related to the CMDB classes, class models, data populating, discovery patterns, and data reconciliation process.

CMDB schema model

A collection of class diagrams and class attributes for key CMDB classes.

CMDB tables descriptions

Descriptions of key CMDB tables in the base system.

CMDB CI Class Models

A ServiceNow Store app that adds class models that extend the base CMDB class hierarchy. The hierarchy includes class descriptions, identification rules, identifier entries, and dependent relationships, if applicable. You can then use the added classes as any other CMDB base class.

Populating the CMDB

Information about the various options for populating the CMDB.

Discovery patterns

A ServiceNow Store app that provides a library of discovery patterns for discovering specific devices and applications in the industry.

CMDB Identification and Reconciliation

A centralized framework for identifying and reconciling data from different data sources that helps maintain the integrity of the CMDB when multiple data sources are used to create and update configuration item (CI) records.

Integration Commons for CMDB (2.11.1)

The Integration Commons for CMDB (sn_cmdb_int_util) store app contains the CMDB Integrations Dashboard and a set of Robust Transform Engine (RTE) transforms and script includes.

Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

Using the CMDB Integrations Dashboard

The Integration Commons for CMDB store app provides a dashboard with a central view of the status, processing results, and processing errors of all installed Service Graph Connectors and any custom integrations created in IntegrationHub ETL.

See [CMDB Integrations Dashboard](#) for more information.

Using RTE transforms as templated operations

The Integration Commons for CMDB (com.snc.cmdb.integration_util) plugin provides the Integration Commons functionality. You can use the transforms and script includes to standardize the values stored in the CMDB by different data integrations or by changes. The attributes that are included in the Integration Commons for CMDB store app are attributes that the Identification and Reconciliation Engine (IRE) requires for identification or attributes that could be used to derive classes.

The transforms are templated operations, meaning that there's a script that controls the logic for the transform. The result is that there can be only a single output. When a transform returns multiple values, then those values are concatenated by a triple pipe (|||). You then must use the split transform to retrieve the values that you're interested in. The inputs are either a single field or a list of fields. For all but one transform, the inputs are assumed to be a fixed list of fields as described for each of the following individual transform.

Note: The RTE transforms are included in the Integration Commons for CMDB store app and are available in the [IntegrationHub ETL \(3.2\)](#) store app. For more information on RTE transforms as templated operations, see [RTE transforms as templated operations](#).

ServiceNow Service Graph Connectors that are available at the ServiceNow Store, have dependencies on the transforms and script includes in the Integration Commons for CMDB store app. Therefore, when you install such CMDB integration, the Integration Commons for CMDB store app is automatically installed too.

You can also configure the Application Dependency Mapping (ADM) adapter to populate running processes, TCP connections, and applications into CMDB. For more information, see [Configuring the ADM adapter](#).

Important: After upgrades and deployments of new applications or integrations, run quick start tests to verify that Integration Commons for CMDB works as expected. See [Quick start tests](#) for more information.

- [CMDB Integrations Dashboard](#)

You can use the CMDB Integrations Dashboard that provides a central view of status, processing results, and processing errors of all installed Service Graph Connectors and any custom integrations created in IntegrationHub ETL run.

- [RTE transforms included within the Integration Commons for CMDB app](#)

The Robust Transform Engine (RTE) transforms are templated operations included within the Integration Commons for CMDB (`sn_cmdb_int_util`) store app.

- [Configuring the ADM adapter for Service Graph Connectors](#)

You can configure the Application Dependency Mapping (ADM) adapter to populate running processes, TCP connections, and applications into CMDB.

- [Quick start tests for Integration Commons for CMDB](#)

Validate that integrations for CMDB pass validation and still work after you make any configuration changes such as applying an upgrade or developing an application.

Related concepts

- [IntegrationHub ETL \(3.2\)](#)
- [Service Graph Connectors](#)
- [Available Service Graph Connectors](#)
- [Service Graph Connector for AWS \(2.2.1\)](#)
- [Service Graph Connector for Microsoft Azure \(1.4.0\)](#)
- [Service Graph Connector for Observability - AppDynamics \(1.2.1\)](#)

- Service Graph Connector for Observability - Datadog (1.2.1)
- Service Graph Connector for Observability - Dynatrace (1.8.0)
- Service Graph Connector for Observability - New Relic (1.2.1)
- Service Graph Connector for ExtraHop (2.0.3)
- Service Graph Connector for GCP (1.3.1)
- Service Graph Connector for Infoblox (1.1.0)
- Service Graph Connector for Microsoft Intune (2.3.0)
- Service Graph Connector for Jamf (2.12.0)
- Service Graph Connector for Microsoft SCCM (3.4.0)
- Service Graph Connector for OpenTelemetry (1.2.0)
- Service Graph Connector for SolarWinds (2.4.1)
- Service Graph Connector for Tanium (1.5.0)
- Service Graph Connector for VMware Workspace ONE UEM (1.6.0)

You can use the CMDB Integrations Dashboard that provides a central view of status, processing results, and processing errors of all installed Service Graph Connectors and any custom integrations created in IntegrationHub ETL run.

The CMDB Integrations Dashboard is available with the Integration Commons for CMDB (sn_cmdb_int_util) store app. On the dashboard, you can see metrics for all Service Graph Connector runs, or filter the view to a specific connector, a specific time duration, or a specific connector run.

Learn more about the CMDB Integrations Dashboard from the following video.

Access CMDB Integrations Dashboard

Access the CMDB Integrations Dashboard provided with the Integration Commons for CMDB store app for maintaining data consistency and accuracy across multiple data sources.

Before you begin

Role required: None

Procedure

1. Navigate to **Self-Service > Dashboards**.
2. On the Dashboards view, select **CMDB Integrations Dashboard** and do any of the following actions:
 - Select the **CMDB Execution Status** tab to see metrics such as the total number of integrations and processed rows, integration runs actively running, daily statistics, and details about the classes that were updated.

- Select the **CMDB Integration Errors** tab to see metrics such as number of import and integration errors, and number of erroneous imported records.
- Point to the score on the various tiles to drill down to the list views for the associated records. Point to the charts to show more details for the chart.
- Narrow down the scope of the integration runs included in the metrics on the dashboard by configuring filters on the right-hand side of the dashboard. Set any of the following filters and then select **Apply**. The filter settings apply to any metric with a filter icon in its upper left corner.

Filters

Filter	Description
Import Date	Select All or a time period, such as Last 7 days , from which to include integration runs in metrics.
CMDB Applications	Select All , or a CMDB application, such as SCCM , or a custom integration, from which to include integration runs in metrics.
CMDB Import	Select All , or a specific integration run to include in metrics.

The Robust Transform Engine (RTE) transforms are templated operations included within the Integration Commons for CMDB (`sn_cmdb_int_util`) store app.

The following RTE transforms are available in the Integration Commons for CMDB app.

CI Lookup Operation

Use to get the value of a field on an existing configuration item (CI) in the CMDB by the source native key.

Details	
Table	sn_cmdb_int_util_ci_lookup_operation
Input field	<p>source_sys_rte_eb_field Input in order is:</p> <ol style="list-style-type: none"> 1. Discovery Source 2. Source Native Key 3. CI Field <p>The operation queries the Source [sys_object_source] table for the discovery source and the associated source native key, and then returns the CI Field value of the matching record in the target table and the associated target sys ID.</p>
Output field	<p>target_sys_rte_eb_field</p> <p>Output is the value of the field name on the CI matched by the source native key lookup or an empty string if there is no match.</p>

The Source [sys_object_source] table is queried using the discovery source and source native key ordering by the last scan. The table iterates through the results of the query and queries the target table by the target sys ID until a valid CI is found. After a valid CI is found, the operation returns the value of the CI Field on the matching CI.

Example

Discovery Source	Source Native Key	CI Field	Result
ServiceNow	ServiceNow COMPUTER-NAME Computer-01	name	Computer-01

Cleanse Company

Use to cleanse hardware manufacturer name and add the record to the Company [core_company] table to populate a reference, when the manufacturer is not linked to a model or software (cpu_manufacturer).

Details	
Table	sn_cmdb_int_util_cleanse_company_operation
Input field	source_sys_rte_eb_field Input is a company/manufacturer name.
Output field	target_sys_rte_eb_field Output is the resulting sys_id and name of the company in core_company, concatenated by triple pipe ().
Script include function	sn_cmdb_int_util.CmdbIntegrationHardwareModelUtil().cleanseCompany(input)

If a matching record does not exist, then a new record is created in core_company so the return always includes a sys_id and name (unless the input is empty or invalid). The name is cleansed and a fuzzy lookup is done via the CmdbIntegrationCompanyModelUtil script include before the MakeAndModelJS platform API is called.

Example

Input	Result
SERVICENOW	93d4ecfac0a8000b6294d71b73397 7fb ServiceNow

Cleanse Hardware Model

Use to create, cleanse, or lookup a hardware model to create a reference (model_id).

Details	
Table	sn_cmdb_int_util_cleanse_hardware_model_operation
Input fields	<p>source_sys_rte_eb_fields Input in order is:</p> <ol style="list-style-type: none">1. The manufacturer name2. The model name <p>If either value is provided by itself, then the operation only processes what is found.</p>
Output field	<p>target_sys_rte_eb_field</p> <p>Output is the resulting sys_id and name of the company in core_company, and sys_id and name of the model in cmdb_model - all concatenated by triple pipe ().</p>
Script include function	<pre>sn_cmdb_int_util.CmdbIntegrationHardwareModelUtil().cleanseModelAndCompany(manufacturer_in, model_in)</pre>

For either the manufacturer or model, if a matching record does not exist then a new record is created so the return always includes sys_ids and names for both records (unless the input is empty or invalid).

The manufacturer name is processed like the Cleanse Company transform and then the manufacturer name and model name are sent to the MakeAndModelJS platform API.

Example

Manufacturer Name	Model Name	Result
ServiceNow Incorporated	SERVICENOW	93d4ecfac0a8000b629 4d71b733977fb ServiceNow ba29cb303710200044e 0bfc8bcbe5d6d ServiceNow

Cleanse Hardware Model with Model Number

Use to create, cleanse, or lookup a hardware model to create a reference (model_id).

Details	
Table	sn_cmdb_int_util_cleanse_hardware_model_operation
Input fields	<p>source_sys_rte_eb_fields Input in order is:</p> <ol style="list-style-type: none"> 1. The manufacturer name 2. The model name 3. The model number 4. CI Class <p>Note: The CI Class field is an optional input field, and when included, the MakeAndModelJS platform API gets the model record by querying the product model class from the Model Category [cmdb_model_category] table. Else, the API gets the model record from the Hardware Models [cmdb.hardware_product_model] table only.</p>

Details	
	If either value is provided by itself, then the operation only processes what is found.
Output field	<p>target_sys_rte_eb_field</p> <p>Output is the resulting sys_id and name of the company in core_company, and sys_id and name of the model in cmdb_model - all concatenated by triple pipe ().</p>
Script include function	sn_cmdb_int_util.CmdbIntegrationHardwareModelUtil().cleanseModelAndCompany(manufacturer_in, model_in)

For either the manufacturer or model, if a matching record does not exist then a new record is created so the return always includes sys_ids and names for both records (unless the input is empty or invalid).

The manufacturer name is processed like the Cleanse Company transform and then the manufacturer name, model name, and model number are sent to the MakeAndModelJS platform API.

Example

Manufacturer Name	Model Name	Model Number	Result
ServiceNow Incorporated	SERVICENOW	BC0AA8000C56	93d4ecfac0a80 00b6294d71b733 977fb ServiceNow ba29cb3037102 00044e0bfc8bc be5d6d ServiceNow

Cleanse IP Address

Use when a field provides an IP address.

Details	
Table	sn_cmdb_int_util_cleanse_ip_operation
Input fields	source_sys_rte_eb_field Input is the IP address to cleanse.
Output field	target_sys_rte_eb_field Output is the resulting IP address, which can be empty.
Script include function	sn_cmdb_int_util.CmdbIntegrationNetworkUtil().cleanseIpAddress(input)

The IP address is tested for both IPv4 and IPv6 structures along with some known derivations (an IPv4 with spaces instead of periods). If a result is found, then it is formatted and returned.

Examples

Ip	Ip Results
192.160.89.1	192.160.89.1
192.160.89.1,54.21.12.311	192.160.89.1
192 160 89 1 54 21 12 311	192.160.89.1
192-160-89-1	192.160.89.1
2001:0db8:0000:0000:0000:ff00:0042 :8329	2001:0db8:0000:0000:0000:ff00:0042 :8329
junk	
175912537	10.124.54.89

Ip	Ip Results
-1	
0	

Cleanse IP Version

Use when the source of data does not provide an IP version or when the IP version might be unreliable.

Details	
Table	sn_cmdb_int_util_cleanse_ip_version_operation
Input fields	source_sys_rte_eb_field Input is the IP address to cleanse.
Output field	target_sys_rte_eb_field Output is the resulting cmdb_ci_ip_address.ip_version lookup key (either 4, 6, or empty).
Script include function	sn_cmdb_int_util.CmdbIntegrationNetworkUtil().derivelpVersion(input)

The input IP address value is checked for either proper IPv4 or IPv6 structure, otherwise the return is empty. This function provides no IP cleansing.

Example

Input	Result
192.160.89.1	4

Cleanse MAC Address

Use when a field provides a MAC address.

Details	
Table	sn_cmdb_int_util_cleanse_mac_operation
Input fields	source_sys_rte_eb_field Input is the MAC address to cleanse.
Output field	target_sys_rte_eb_field Output is the resulting MAC address which can be empty.
Script include function	sn_cmdb_int_util.CmdbIntegrationNetworkUtil().cleanseMacAddress(input)

The MAC address is tested for proper structure along with some known derivations (for example, a MAC address with spaces instead of colons). If a result is found, then it is formatted and returned.

Example

Input	Result
00 0A 95 9D 68 16	00:0a:95:9d:68:16

Cleanse Operating System

Use to extract, cleanse, and format an operating system name, when the source provides an operating system value.

Details	
Table	sn_cmdb_int_util_cleanse_os_operation
Input fields	<p>source_sys_rte_eb_field Input is the operating system name to cleanse.</p>
Output field	<p>target_sys_rte_eb_field Output is the resulting operating system name. The resulting operating system name is also written to the cmdb_ci_computer.os list field.</p>
Script include function	sn_cmdb_int_util.CmdbIntegrationOsUtil().cleanseAndInserOs(input)

Most of the current cleansing is centered on Microsoft operating system values aside from common cleansing such as fixing casing.

Examples

Operating System	Operating System Results
Windows Server 2003 R2 64 bit Edition Service Pack 2	Windows Server 2003 R2
Windows 2003	Windows 2003
Windows Vista 64 bit Edition	Windows Vista
Windows 2000 Professional Service Pack 4	Windows 2000 Professional
Windows XP Service Pack 2-3	Windows XP
Microsoft Windows Server 2003 R2 64 bit Edition Service Pack 2	Windows Server 2003 R2

Operating System	Operating System Results
Microsoft Windows 2003	Windows 2003
Microsoft Windows Vista 64 bit Edition	Windows Vista
Microsoft Windows 2000 Professional Service Pack 4	Windows 2000 Professional
Microsoft Windows XP Service Pack 2-3	Windows XP
linux ubuntu	Linux Ubuntu
Linux Ubuntu Server	Linux Ubuntu Server

Cleanse Serial Number

Use to cleanse and remove invalid serial numbers.

Details	
Table	sn_cmdb_int_util_cleanse_serial_number_operation
Input fields	source_sys_rte_eb_fields Input is the serial number to cleanse.
Output field	target_sys_rte_eb_field Output is the resulting serial number.

Examples

Serial Number	Serial Number Results
ec2aa2da-5312-aa3e-804c-c35feabeda5f	ec2aa2da-5312-aa3e-804c-c35feabeda5f

Serial Number	Serial Number Results
1045-1209-6738-4668-7696-2783	1045-1209-6738-4668-7696-2783

Cleanse Software Model

Use to cleanse and create a software model. Also, to create manufacturer and software model if they do not exist and follow with a split operation.

Details	
Table	sn_cmdb_int_util_cleanse_software_model_operation
Input fields	<p>source_sys_rte_eb_fields Input in order is:</p> <ol style="list-style-type: none">1. The manufacturer name2. The software name3. The software version (not required) <p>If only manufacturer or name is provided, then only those values are processed and returned.</p>
Output field	<p>target_sys_rte_eb_field</p> <p>Output is the resulting sys_id and name of the company in core_company, the cleansed software name, and the cleansed software version all concatenated by a triple pipe ().</p>
Script include function	sn_cmdb_int_util.CmdbIntegrationSoftwareModelUtil().cleanseSoftwareModel(company, model, version)

If a matching manufacturer record does not exist, then a new record is created so the return always includes the sys_id and name for the manufacturer (if the manufacturer is not empty or invalid).

The manufacturer name is processed the same as in the Cleanse Company transform and then the manufacturer name is sent to the MakeAndModelJS platform API.

The software name and version are cleansed and formatted and returned. The version is removed from the software name if present.

Examples

Manufacturer	Software Name	Software Version	Results
Dell Inc.			
	NoManufacturer		
		1.0.0.0	
Dell Inc.	DataEngine	1.0.17.2	b7e7d7d8c0a8016900a5d7f 291acce5c Dell Inc. DataEngine 1.0.17.2
GenuineIntel	TestSoftware	1.0.0.1	7aad6d00c611228400f00e0 f80b67d2d Intel TestSoftware 1.0.0.1
Dell Inc.	TestSoftware	232	b7e7d7d8c0a8016900a5d7f 291acce5c Dell Inc. TestSoftware 232
Dell Inc.	TestSoftware	123.0.0.0	b7e7d7d8c0a8016900a5d7f 291acce5c Dell Inc. TestSoftware 123.0
America Online	TestSoftware	1.0.0.0	0c43d035c61122750000251 553f6f8e8 America Online TestSoftware 1.0
America Online	TestSoftware	1.0.0.0	0c43d035c61122750000251 553f6f8e8 America

Manufacturer	Software Name	Software Version	Results
			Online TestSoftware 1.0
dell	LowerCase	1.0.0.0	b7e7d7d8c0a8016900a5d7f291acce5c Dell Inc. LowerCase 1.0
Dell Corporation. Incorporate d, Corp.	TestSoftware	1.0.0.0	b7e7d7d8c0a8016900a5d7f291acce5c Dell Inc. TestSoftware 1.0
Microsoft	Microsoft SQL Server 2016 Enterprise	2.0.0	0e8b8e650a0a0b3b004f285ffbb1a4fc Microsoft Microsoft SQL Server 2016 Enterprise 2.0
Dell Computer	DataEngine	1.0.17.2	b7e7d7d8c0a8016900a5d7f291acce5c Dell Inc. DataEngine 1.0.17.2
Adobe	TestSoftware	1	b7e8b5c4c0a80169008b49e468920048 Adobe Systems TestSoftware 1.0

Create Software Instance Name

Use when a hardware name, software name, and software version is provided, to create a new software instance name.

Details	
Table	sn_cmdb_int_util_create_software_instance_name_operation
Input fields	<p>source_sys_rte_eb_fields Input in order is:</p> <ol style="list-style-type: none"> 1. The hardware name 2. The software name 3. The software version (not required)
Output field	<p>target_sys_rte_eb_field Output is the software instance name.</p>
Script include function	sn_cmdb_int_util.CmdbIntegrationSoftwareModelUtil().createSoftwareInstanceName(hw_name_in, sw_name_in, sw_version_in)

Examples

Hardware Name	Software Name	Software Version	Results
computer1	microsoft	2.0.1	microsoft 2.0.1-computer1
computer2	adobe		adobe-computer2
computer3	adobe	2.1	adobe 2.1-computer3
hw2	sw3	301	sw3 301-hw2

Derive CI Class from Model

Use when processing a computer record and a model is provided but the class of the computer is ambiguous otherwise. Can be used along with other Derive CI Class transforms.

Details	
Table	sn_cmdb_int_util_derive_class_from_model_operation
Input fields	source_sys_rte_eb_fields Input in order is: <ol style="list-style-type: none">1. The model name2. The current class name
Output field	target_sys_rte_eb_field Output is the resulting class name.
Script include function	sn_cmdb_int_util.CmdbIntegrationClassUtil().deriveClassNameFromModelInput(model_in, class_in)

Does not return a value of a class which is higher in the class hierarchy (a parent class) than the provided input class. For example, does not return cmdb_ci_computer if the input is cmdb_ci_server. Looks only at the cmdb_ci_computer hierarchy, going through cmdb_ci_server (cmdb_ci_computer, cmdb_ci_server, children of cmdb_ci_server).

Currently looks for Server, Windows Server, and Linux Server indicators in the model.

Examples

Model	Class	Class Results
window server	cmdb_ci_computer	cmdb_ci_win_server

Model	Class	Class Results
Microsoft server	cmdb_ci_computer	cmdb_ci_win_server
linux server	cmdb_ci_computer	cmdb_ci_linux_server
Microsoft server	cmdb_ci_server	cmdb_ci_win_server
linux server	cmdb_ci_server	cmdb_ci_linux_server
Red hat server	cmdb_ci_server	cmdb_ci_linux_server
Arch server	cmdb_ci_server	cmdb_ci_linux_server
Centos server	cmdb_ci_server	cmdb_ci_linux_server
Debian server	cmdb_ci_server	cmdb_ci_linux_server
Fedora server	cmdb_ci_server	cmdb_ci_linux_server
Suse server	cmdb_ci_server	cmdb_ci_linux_server
Oracle server	cmdb_ci_server	cmdb_ci_linux_server
Rhel server	cmdb_ci_server	cmdb_ci_linux_server
Ubuntu server	cmdb_ci_server	cmdb_ci_linux_server
Junk		
	cmdb_ci_server	cmdb_ci_server
Junk server	cmdb_ci_computer	cmdb_ci_server
Junk	cmdb_ci_computer	cmdb_ci_computer

Derive CI Class from Native Class Identifier

Use when processing a computer record and a native class indicator is provided but the class of the computer is ambiguous otherwise. Can be used along with other Derive CI Class transforms.

Details	
Table	sn_cmdb_int_util_derive_class_from_native_value_operation
Input fields	source_sys_rte_eb_fields Input in order is: 1. The native class identifier 2. The current class name
Output field	target_sys_rte_eb_field Output is the resulting class name.
Script include function	sn_cmdb_int_util.CmdbIntegrationClassUtil().deriveClassNameFromNativeValue(native_id_in, class_in)

Does not return a value of a class which is higher in the class hierarchy (a parent class) than the provided input class. For example, does not return cmdb_ci_computer if the input is cmdb_ci_server. Looks only at the cmdb_ci_computer hierarchy, going through cmdb_ci_server (cmdb_ci_computer, cmdb_ci_server, children of cmdb_ci_server).

Currently looks for Server, Windows Server, and Linux Server indicators in the native identifier.

Examples

Native Class	Class	Class Results
window server	cmdb_ci_computer	cmdb_ci_win_server
Microsoft server	cmdb_ci_computer	cmdb_ci_win_server
linux server	cmdb_ci_computer	cmdb_ci_linux_server

Native Class	Class	Class Results
Microsoft server	cmdb_ci_server	cmdb_ci_win_server
linux server	cmdb_ci_server	cmdb_ci_linux_server
Red hat server	cmdb_ci_server	cmdb_ci_linux_server
Arch server	cmdb_ci_server	cmdb_ci_linux_server
Centos server	cmdb_ci_server	cmdb_ci_linux_server
Debian server	cmdb_ci_server	cmdb_ci_linux_server
Fedora server	cmdb_ci_server	cmdb_ci_linux_server
Suse server	cmdb_ci_server	cmdb_ci_linux_server
Oracle server	cmdb_ci_server	cmdb_ci_linux_server
Rhel server	cmdb_ci_server	cmdb_ci_linux_server
Ubuntu server	cmdb_ci_server	cmdb_ci_linux_server
Junk		
	cmdb_ci_server	cmdb_ci_server
Junk server	cmdb_ci_computer	cmdb_ci_server
Junk	cmdb_ci_computer	cmdb_ci_computer

Derive CI Class from Operating System

Use when processing a computer record and an operating system is provided but the class of the computer is ambiguous otherwise. Can be used along with other Derive CI Class transforms.

Details	
Table	sn_cmdb_int_util_derive_class_from_os_operation
Input fields	<p>source_sys_rte_eb_fields Input in order is:</p> <ol style="list-style-type: none"> 1. The operating system name 2. The current class name
Output field	<p>target_sys_rte_eb_field Output is the resulting class name.</p>
Script include function	sn_cmdb_int_util.CmdbIntegrationClassUtil().deriveClassNameFromOsName(os_in, class_in)

Does not return a value of a class which is higher in the class hierarchy (a parent class) than the provided input class. For example, does not return cmdb_ci_computer if the input is cmdb_ci_server. Looks only at the cmdb_ci_computer hierarchy, going through cmdb_ci_server (cmdb_ci_computer, cmdb_ci_server, children of cmdb_ci_server).

Currently looks for Server, Windows Server, and Linux Server indicators in the operating system name.

Examples

Operating System	Class	Class Results
window server	cmdb_ci_computer	cmdb_ci_win_server
Microsoft server	cmdb_ci_computer	cmdb_ci_win_server
linux server	cmdb_ci_computer	cmdb_ci_linux_server
Microsoft server	cmdb_ci_server	cmdb_ci_win_server

Operating System	Class	Class Results
linux server	cmdb_ci_server	cmdb_ci_linux_server
Red hat server	cmdb_ci_server	cmdb_ci_linux_server
Arch server	cmdb_ci_server	cmdb_ci_linux_server
Centos server	cmdb_ci_server	cmdb_ci_linux_server
Debian server	cmdb_ci_server	cmdb_ci_linux_server
Fedora server	cmdb_ci_server	cmdb_ci_linux_server
Suse server	cmdb_ci_server	cmdb_ci_linux_server
Oracle server	cmdb_ci_server	cmdb_ci_linux_server
Rhel server	cmdb_ci_server	cmdb_ci_linux_server
Ubuntu server	cmdb_ci_server	cmdb_ci_linux_server
Junk		
	cmdb_ci_server	cmdb_ci_server
Junk server	cmdb_ci_computer	cmdb_ci_server
Junk	cmdb_ci_computer	cmdb_ci_computer

Derive Virtual From Hardware Model

Use when processing a computer record that may be virtual, a hardware model is provided, and the virtual status is ambiguous. Can be used along with other Derive Virtual From transforms.

Details	
Table	sn_cmdb_int_util_derive_virtual_from_model_operation

Details	
Input fields	source_sys_rte_eb_fields Input in order is: 1. The hardware model name 2. The current virtual flag value
Output field	target_sys_rte_eb_field Output is the resulting virtual flag (true/false). If the current virtual flag is 'true', the result is true. Otherwise the result is 'true' or 'false'.
Script include function	<pre>sn_cmdb_int_util.CmdbIntegrationVirtualDetectionUtil().detectVirtualFromModelName(model_in, is_virtual_in)</pre>

Looks for indicators in the model name for a virtual device (VMware).

Examples

Hardware Model	Virtual Flag	Virtual Flag Results
thinkpad	true	true
thinkpad	false	false
thinkpad		false
vmware inc	true	true
	true	true
	false	false
		false

Derive Virtual From Native Indicator

Use when processing a computer record that may be virtual, a virtual indicator is provided by the source, and the virtual status is ambiguous. Can be used along with other Derive Virtual From transforms.

Details	
Table	sn_cmdb_int_util_derive_virtual_from_native_value_operation
Input fields	source_sys_rte_eb_fields Input in order is: <ol style="list-style-type: none">1. The native indicator2. The current virtual flag value
Output field	target_sys_rte_eb_field Output is the resulting virtual flag (true/false). If the current virtual flag is 'true', the result is true. Otherwise the result is 'true' or 'false'.
Script include function	sn_cmdb_int_util.CmdbIntegrationVirtualDetectionUtil().detectVirtualFromNativeIdentifier(native_in, is_virtual_in)

Tests native indicator against a list of common values and looks for a 'true' boolean indicator.

Examples

Native Virtual Value	Virtual Flag	Virtual Flag Results
virtual	false	true
virtual		true

Native Virtual Value	Virtual Flag	Virtual Flag Results
virtual	true	true
y	false	true
y		true
y	true	true
yes	false	true
yes		true
yes	true	true
true	false	true
true		true
true	true	true
t	false	true
t		true
t	true	true
other	false	false
other		false
other	true	true
not virtual	false	false
not virtual		false
not virtual	true	true
	false	false

Native Virtual Value	Virtual Flag	Virtual Flag Results
		false
	true	true

Derive Virtual From Serial Number

Use when processing a computer record that may be virtual, a serial number is provided by the source, and the virtual status is ambiguous. Can be used along with other Derive Virtual From transforms.

Details	
Table	sn_cmdb_int_util_derive_virtual_from_serial_number_operation
Input fields	source_sys_rte_eb_fields Input in order is: 1. The serial number 2. The current virtual flag value
Output field	target_sys_rte_eb_field Output is the resulting virtual flag (true/false). If the current virtual flag is 'true', the result is true. Otherwise the result is 'true' or 'false'.
Script include function	sn_cmdb_int_util.CmdbIntegrationVirtualDetectionUtil().detectVirtualFromSerialNumber (serial_in, is_virtual_in)

Looks for indicators in the serial number for a virtual device (VMware).

Examples

Serial Number	Virtual Flag	Virtual Flag Results
123	true	true
123	false	false
123		false
vmware-123	true	true
	true	true
	false	false
		false

Extract and Scale by Units

Use when the source has numerical values that need to be scaled and numerical value with an input such as 2048Mb. The source does not always provide the units so it may be required to calculate or guess the units being provided. The target units depend on the target field in the CMDB. If not specified, the decimal place field is set at 2 by default.

Note: This field is case sensitive.

Details	
Table	sn_cmdb_int_util_extract_and_scale_by_units_operation
Input fields	source_sys_rte_eb_fields
Output field	target_sys_rte_eb_field
Script include function	sn_cmdb_int_util.CmdbIntegrationExtractScaleUnitUtil().extractAndScaleUnits(input,defaultUnit,outputUnit,decimalPlaces)

Examples

Input Value	Default Unit	Output Unit	Result
2048Mb	Mb	GB	2GB
17179869184	B	GB	16GB

First Non Null Value

Use when you have a list of fields providing similar information that must map to a single field and you want to rank the order in which they can provide those values.

For Example Internally in SolarWinds, there is a hierarchy of tables that are join. In one example, a computer's name could come from the child most table or any of that table's parents but each of those is a separate field in the pull. Starting with the most specific table, the values are searched for the first instance of a name value.

Details	
Table	sn_cmdb_int_util_first_non_null_operation
Input fields	source_sys_rte_eb_fields Input is a list of fields of any length.
Output field	target_sys_rte_eb_field Output is the value from the first field in the list that doesn't have a null (or empty) value.
Script include function	sn_cmdb_int_util.CmdbIntegrationFirstNonNullValueUtil().firstNonNullValue(batch[i])

Example

Field 1	Field 2	Field 3	Result
	foo	foo2	foo

Process Name, Domain, FQDN, DNS set

Use when the source provides name, domain, FQDN, or DNS information. Can be used for only a subset of these (if for example, the source only provides name and domain). In the case that a source only provides fields that are lower in the input list (FQDN) the CmdblIntegrationHardwareNameUtil script include can be called from a script operation to minimize having to create empty dummy fields.

Details	
Table	sn_cmdb_int_util_process_name_set_operation
Input fields	<p>source_sys_rte_eb_fields Takes up to four Input fieldss (any additional fields are ignored), in the following order:</p> <ol style="list-style-type: none">1. Name2. Domain3. FQDN4. DNS <p>You don't have to provide all four input values, but you must provide those values in the specified order. If for example, you only want to cleanse domain, you must provide a name attribute, even it if empty.</p>
Output field	target_sys_rte_eb_field

Details	
	Output is a concatenated set of values in the same order, using a triple pipe (): {name} {domain} {fqdn} {dns}
Script include function	sn_cmdb_int_util.CmdbIntegrationHardwareNameUtil().processNameDomainFqdnDnsSet(name, domain, fqdn, dns)

FQDN and DNS are first processed to see if their formats are correct. FQDN has an additional discovery regex it must pass (via properties):
`glide.discovery.fqdn.regex - default : ^([^.]+)\.\.(?:[^.]+\\.)+[^.]+)$`

Possible name and domain values are extracted if possible. When name and domain are processed, if there is no FQDN, a value is generated if possible. A resulting name value is also modified using the following discovery flags:

1. glide.discovery.hostname.case – default: No change. Can be set to ‘Lower case’, ‘Upper case’, ‘No change’
2. glide.discovery.hostname.include_domain – default: false. If ‘true’ the domain is added to the final name value

Examples

Name	Domain	FQDN	DNS	Results
myName	other.net	otherName.other.net	mycomp.servicenow.com	myName other.net otherName.other.net mycomp.servicenow.com

Name	Domain	FQDN	DNS	Results
na	other.net	otherName.other.net	mycomp.servicenow.com	otherName other.net otherName.other.net mycomp.servicenow.com
			servicenow.com	servicenow.com
			name.servicenow.com	name servicenow.com name.servicenow.com name.servicenow.com
		name.servicenow.com		name servicenow.com name.servicenow.com
		name.servicenow.com		name servicenow.com name.servicenow.com
name	servicenow.com			name servicenow.com

Name	Domain	FQDN	DNS	Results
				name.servicenow.com

Process FQDN

Use when the source provides a suspected FQDN value but no other naming fields such as name, domain, or DNS.

Details	
Table	sn_cmdb_int_util_process_fqdn_operation
Input fields	source_sys_rte_eb_field Input is a single field containing an FQDN.
Output field	target_sys_rte_eb_field Output is a concatenated set of values in the same order using a triple pipe (): {name} {domain} {fqdn} {dns}
Script include function	sn_cmdb_int_util.CmdbIntegrationHardwareNameUtil().processNameDomainFqdnDnsSet("", "", fqdn, "")

The processing follows the same logic as the 'Process Name, Domain, FQDN, DNS set' transform except that only FQDN is used as an input.

Example

Input	Result
mycomputer.servicenow.com	mycomputer servicenow.com mycomputer.servicenow.com

Scale Units

Use when the source has numerical inputs that must be scaled. The source does not always provide the current units so it may be required to calculate or guess the units being provided. The target units depend on the field being targeted in the CMDB.

Details	
Table	sn_cmdb_int_util_scale_unit_operation
Input fields	source_sys_rte_eb_fields Inputs in order are: <ol style="list-style-type: none">1. The input value2. The current units3. The target units
Output field	target_sys_rte_eb_field Output is the input value scaled from the current units to the target units. If no units are found for the current units, then the input value is returned. If no current or target units are found the input is returned as the output.
Script include function	sn_cmdb_int_util.CmdbIntegrationScaleUnitUtil().scaleUnits(input_value, input_unit, output_unit)

Example

Input	Result
<ul style="list-style-type: none">• Input Field 1: 1• Input Field 2: GB	1024

Input	Result
• Input Field 3: MB	

Software Bundle ID Lookup

Use when a source, such as Jamf, does not provide the software publisher but does provide a Mac software bundle ID. Software Bundle ID Lookup looks up records in the Bundleid Lookup [sn_cmdb_int_util_bundleid_lookup] table by bundle_id. If a record with the specified bundle_id exists, it extracts the respective software publisher. Otherwise, it creates a new record which will be queried the next time the Lookup Mac Software Bundle IDs data source runs.

Details	
Table	sn_cmdb_int_util_software_bundle_id_lookup_operation
Input fields	<ul style="list-style-type: none">source_sys_rte_eb_fieldBundle ID
Output field	target_sys_rte_eb_field Output is the resulting artist name, track name, and seller name, all concatenated by a triple pipe (), or an empty string if no match is found.
Script include function	sn_cmdb_int_util.CmdbIntegrationSoftwareBundleIdLookup.lookupSoftware(bundleId)

Example

Input	Result
Input Field 1: com.microsoft.Word	Microsoft Corporation Microsoft Word Microsoft Corporation

User Lookup

Use to look up a user in the User [sys_user] table by user name or email, attempting to match in the following order:

1. The User Name matching the user_name attribute.
2. The Email matching the email attribute.
3. If nothing is matching, it returns empty.

Details	
Table	sn_cmdb_int_util_user_lookup_operation
Input fields	source_sys_rte_eb_fields Inputs in order are: 1. User Name 2. Email (Optional)
Output field	target_sys_rte_eb_field sysId of the sys_user.
Script include function	sn_cmdb_int_util.CmdbIntegrationUserLookup.lookupUser(username, email)

Examples

Input	Result
Input Field 1: abel.tuter	62826bf03710200044e0bfc8bcbe5 df1
• Input Field 1: atuter	62826bf03710200044e0bfc8bcbe5 df1

Input	Result
• Input Field 2: abel.tuter@example.com	

You can configure the Application Dependency Mapping (ADM) adapter to populate running processes, TCP connections, and applications into CMDB.

As a user with the admin role, you can use the ADMHelper script include to configure the ADM adapter that populates running processes, TCP connections, and applications into CMDB. The ADMHelper script include is available within the Integration Commons for CMDB (sn_cmdb_int_util) store app. The ADMHelper script include invokes the ApplicationDependencyMapping script include that is available within the Discovery and Service Mapping Patterns application (sn_itom_pattern).

The ADM adapter requires the inputs as discussed in the following table:

Inputs for the ADM adapter

Input	Input type
Running Process Details	Required
Computer Sys Id	Required
TCP connection details	Optional
ADM Properties	Optional

For interpreting and populating the CIs, the ADM processor requires the input data in a specific format. Ensure that the keys are formed as shown in the following example:

Example keys for ADM processor

TCP connections data	Running process
[{ "pid": "1068", "local_ip": "127.0.0.1", "local_port": "199", "ip": "0.0.0.0", "port": "199", "state": "LISTEN", "type": "on" }]	[{ "pid": "1", "ppid": "0", "command": "/usr/lib/systemd/systemd", "name": "systemd", "parameters": "--switched-root --system --deserialize 21" }]

The ApplicationDependencyMapping script include processes TCP connections and running process data and populates the following tables:

- TCP Connections [cmdb_tcp]
- Running Process [cmdb_running_process]
- Application [cmdb_ci_appl]

Note: After the data is populated into the TCP Connections [cmdb_tcp] and Running Process [cmdb_running_process] tables, the ApplicationDependencyMapping script include reconciles the TCP connections and running process data for populating the Application [cmdb_ci_appl] table based on the data in the Network Adapter [cmdb_ci_network_adapter] and IP Address [cmdb_ci_ip_address] tables.

Validate that integrations for CMDB pass validation and still work after you make any configuration changes such as applying an upgrade or developing an application.

Danger: By default, the system property that is used to run automated tests is disabled to prevent you from accidentally running these tests on a production system. To avoid data corruption or an outage, run tests only on development, test, and other non-production instances. See [Enable or disable executing Automated Test Framework tests](#).

Integration Commons for CMDB

CMDB INT: CMDB Integrations Validation test suite

Test suite to verify the integrity of an integration using multiple tests.

Test	Description	Release version
CMDB INT: Set Test Session Application	Modify the run server-side script to set an application name so that you can test only one integration. Otherwise, all integrations installed will be tested.	Paris
CMDB INT: Test Against Source Analysis	Test an integration against the values in the CMDB Integration Source Analysis [sn_cmdb_int_util_cmd_b_integration_source_analysis] table.	Paris
CMDB INT: Validate Application Feed	Validate all application feeds in an integration.	Paris
CMDB INT: Validate Discovery Source	Validate that the discovery source exists.	Paris

Test	Description	Release version
CMDB INT: Validate Entity Mappings	Validate all entity mappings of an integration.	Paris
CMDB INT: Validate Fields	Validate fields for CMDB Integrations.	Paris
CMDB INT: Validate Lookups	Validate CMDB integration lookups.	Paris
CMDB INT: Validate Mandatory Operations	Validate that all integrations for mandatory operations exist for mapped fields.	Paris
CMDB INT: Validate Operations	Validate all operations for an integration.	Paris
CMDB INT: Validate References	Validate CMDB integration references.	Paris
CMDB INT: Validate Related Entries	Validate all related classes against the data dictionary for related entries.	Paris
CMDB INT: Validate Relationships	Validate CMDB integration relationships.	Paris

To learn more about Integration Commons for CMDB, see [Integration Commons for CMDB](#).

IntegrationHub ETL (3.2)

Use the IntegrationHub ETL store app to create and manage ETL transform maps, which integrate third-party data into the CMDB or into non-CMDB tables without compromising the integrity of data. IntegrationHub ETL provides a simplified user interface that guides you

through the integration process end-to-end, including a test integration run of sample data.

The IntegrationHub ETL (sn_int_studio) plugin provides the IntegrationHub ETL functionality.

- Use the CMDB Integrations Dashboard to track progress, results, and errors associated with using custom integrations created in IntegrationHub ETL. The CMDB Integrations Dashboard is included in the [Integration Commons for CMDB](#) store app.
- Watch the [IntegrationHub ETL | Importing resources into the CMDB](#) video for an introduction and walk through of the IntegrationHub ETL tool.

Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

Roles required

Users with the cmdb_inst_admin role can use IntegrationHub ETL to create integrations, or customize a pre-existing integration provided by ServiceNow or a vendor at the ServiceNow Store. A vendor can create a new integration and provide it as an application for anyone to use.

Support for non-CMDB tables

Starting with the Vancouver release, IntegrationHub ETL supports the integration of third-party data into some non-CMDB tables. IntegrationHub ETL supports those non-CMDB tables that are supported by Identification and Reconciliation (IRE). For details about which non-CMDB tables are supported and any needed configuration, see [IRE support for non-CMDB tables](#).

Supported non-CMDB tables are available in IntegrationHub ETL when specifying classes, conditional classes, class associations, and reference sources in mapping definitions. However, there are some differences between using CMDB classes and non-CMDB tables in IntegrationHub ETL:

- Specifying class associations isn't mandatory for non-CMDB tables.
- Adding relationships doesn't apply to non-CMDB tables.
- Class associations for a non-CMDB table is based on a reference field instead of a CMDB relationship.

Note: Although the IntegrationHub ETL user interface and accompanying documentation references CMDB and CMDB elements, most of those references also apply to supported non-CMDB tables.

Process

The two key components that IntegrationHub ETL uses for processing are:

- **Robust Transform Engine (RTE):** Used to transform raw source data that is stored in staging tables, into the data that is mapped and integrated into the CMDB. RTE uses ETL transform maps that were created for the integration during data transformation.
- **Identification and Reconciliation engine (IRE):** Used as a centralized framework for identification and reconciliation processes across different data sources. IRE processes help maintain data integrity in the CMDB and in supported non-CMDB tables.

IntegrationHub ETL uses RTE and IRE which work together to process and integrate data. Data is first imported from a data source, and is then stored in temporary staging tables in Import Sets systems. Using the data in the staging tables and the ETL transform map created by IntegrationHub ETL, RTE creates IRE payloads which are then processed by IRE. IRE applies reconciliation processes to avoid potential problems such as duplicate CIs, ensuring that the CMDB or non-CMDB tables remain healthy, and then integrates the resulting data.

When you create an integration, you import source data, transform data if needed, and select target CMDB classes (or non-CMDB tables) and attributes to map the data to. Eventually, you run an integration test of the sample data, using your settings in the IntegrationHub ETL. You can then preview the integration test results and adjust any settings before scheduling recurring integration runs for large data sets. If you develop and test the ETL transform map on a development instance, then you can test and adjust the configuration before implementation on a production instance.

For example, you can integrate data from SCCM (Microsoft System Center Configuration Manager).

Refer to the community page [IntegrationHub-Extract Tranform Load \(IH-ETL\) is GA in ServiceNow store](#) for an overview of IntegrationHub ETL, including its components and workflow.

Guided Setup

A guided setup organizes all the tasks in the correct order, tracks the completion of tasks, and enforces any task dependencies. Tasks that depend on the completion of other tasks, are enabled or disabled as you step through the tool and complete tasks.

Read-only mode

When opening a Service Graph Connector in which IntegrationHub ETL isn't detecting any incoming data from the data source, the integration is available in read-only mode. In read-only mode you can access all the guided setup tasks on the ETL Transform Map Assistant page. You can examine all the settings and definitions in the integration even though it isn't populated with actual data. However, you can't make any updates to a read-only connection.

Read-only mode is useful for studying an existing connection for the purpose of creating a new connection that is similar to the read-only connection. The read-only mode can also assist in troubleshooting issues with the connection.

IntegrationHub ETL and Import Sets

Using IntegrationHub ETL and ETL transform maps has the following advantages over using Import Sets and transform maps:

- Identification and Reconciliation Engine (IRE) processes are incorporated into the IntegrationHub ETL so all data is automatically processed by IRE as part of the integration. Using Import Sets and transform maps does not provide a simple way to apply IRE processes.
- IntegrationHub ETL uses guided setup which provides guidance and a simple user interface for the entire process of integrating third-party data.

- IntegrationHub ETL includes an integration test for a small data set using the new ETL transform map. This test lets you review the results and adjust configuration settings before scheduling recurring integrations.

Terms

The following terms are associated with the IntegrationHub ETL:

CMDB application

Name of the third-party vendor such as SCCM 2019. A CMDB application has two associated attributes: Name and Discovery Source. When creating a new integration, ensure to configure a discovery source for the CMDB application that you plan to use, before using the IntegrationHub ETL.

Data source

The source feed, such as SCCM 7.0 Computer Identity, where the raw source data is imported from. If you use various REST endpoints for different types of data, then each REST endpoint is associated with its own data source and an ETL transform map.

ETL transform map

The output generated by IntegrationHub ETL. You can integrate third-party data into the CMDB or into non-CMDB tables using an ETL transform map which is configured for the respective integration.

Source data

Original, raw data that have been imported into IntegrationHub ETL. Source data can be used in its original form, or you can transform the data before mapping and integration.

Transform

An operation, that you can apply to a specific data column to transform the data values. For example, to transform the format of the data values. Use transforms to standardize data formats and meet other system requirements.

Transformed data

Some of the source data might not be compliant with the requirements of its target CMDB attributes and classes or non-CMDB tables. In those

cases, you can apply various types of transforms to the source data, before mapping the data to the target CMDB classes and attributes or non-CMDB tables. Transforms, can for example convert data format, replace values, and concatenate values from multiple data columns.

Each CMDB application can have multiple connections for retrieving raw data. Each connection that is used to retrieve a certain type of data, has its own pair of data source and an ETL transform map. Therefore, one CMDB application can have multiple ETL transform maps, and each of those ETL transform maps is associated with a single Data Source.

For example:

CMDB Application	ETL Transform Map	Data Source
SCCM	SCCM Computer Identify	/sccm/2019/comp
	SCCM Disk	/sccm/2019/disk
	SCCM Application	/sccm/2019/appl

Nested data payloads

To process nested data payloads, you must first ensure that the data source that is used for the integration, is set with the [Data in single column](#) option. With that setting, you can correctly represent nested data in a JSON payload which IntegrationHub ETL then processes as nested data, rather than as flat data.

Sample of nested data:

```
{  
    "u_computer_fqdn": "computer2-fqdn",  
    "u_computer_id": 2,  
    "u_computer_ip": "computer2-ip",  
    "u_computer_location": "PDX",  
    "u_computer_mac": "computer2-mac",  
    "u_computer_name": "nested-payload-computer2"  
,  
    "u_computer_os": "computer2-os",  
    "interfaces": [  
        {
```

```
        "u_interface_ip": "computer2-eth1-ip"
    ,
        "u_interface_mac": "computer2-eth1-ma
c",
        "u_interface_name": "computer2-eth1",
        "ip": ""
    },
{
    "u_interface_ip": "computer2-eth2-ip"
,
    "u_interface_mac": "computer2-eth2-ma
c",
    "u_interface_name": "computer2-eth2",
    "ip": {
        "u_ip_address": "computer2-eth2-i
p",
        "u_mac_address": "computer2-eth2-
mac"
    }
},
],
"software": [
{
    "u_software_name": "computer2-softwar
e2",
    "u_software_version": "computer2-soft
ware2-1.0",
    "instance": {
        "u_software_instance_name": "comp
uter2-software1-instance"
    }
},
{
    "u_software_name": "computer2-softwar
e2",
    "u_software_version": "computer2-soft
ware2-2.0",
    "instance": {
        "u_software_instance_name": "comp
uter2-software2-instance"
    }
}
]
```

```
    ]  
},
```

You can view the layers of nested data in a separate panel in IntegrationHub ETL, apply transforms, map, and integrate that data into the CMDB.

When creating a nested data JSON payload, the following restrictions apply:

-

Field names must start with a letter (between A-Z or a-z) or with '_', and must only contain letters (between A-Z or a-z), digits (0-9), or the '_' character.

For example, a field name can't contain special characters such as *, [], #, \$, spaces, and dot.

- Field names can't be "temp" or "object", which are reserved for internal use.
- Consistently throughout the payload, you must use an array or an object to represent data in a specific level, regardless of the number of items in the level. If you use an array for multiple items in one object, you must also use an array to represent a single item in other objects.

For a demo about working with nested payload data, watch the [Integration Hub - ETL nested payload feature demo](#) video on the ServiceNow YouTube channel.

Related reference

- [Teams related list](#)

IntegrationHub ETL provides a guided setup which walks you through the completion of all necessary tasks for creating an ETL transform map for a specific integration.

Guided setup

Guided setup organizes all the tasks in the correct order, tracks the completion of tasks, and enforces any task dependencies. Tasks that depend on the completion of other tasks, are enabled or disabled as you step through the tool and complete tasks.

Use guided setup on the ETL Transform Map Assistant page to complete the following tasks.

Import source data and specify basic details

Provide basic details for the integration, such as the source of the data that you want to integrate into CMDB, and import the source data.

Before you begin

The data source that you plan to select for the ETL Transform Map must exist in the same application scope as the one being used in the current session.

When you open an ETL transform map, by default the map is not validated. You can enable this validation step by [adding the system property](#) `sn_int_studio.validation.enabled` to the System Properties [sys_properties] table and then setting it to `true`. After validation is complete, you choose how to handle validation errors.

Role required: `cmdb_inst_admin`

Procedure

1. Navigate to **All > Configuration > IntegrationHub ETL**.

The landing page of the IntegrationHub ETL lists all integrations that exist in the system, including integrations that were downloaded from the ServiceNow Store. Starting with IntegrationHub ETL v3.2, integrations are grouped by the CMDB Application value, in which case expand the respective group to locate an integration.

2. Click the **Name** of an integration to view or modify, or click **Create new**.

If the system property `sn_int_studio.validation.enabled` is set to `true`, then IntegrationHub ETL validates the ETL transform map that you are loading. If there are any validation errors, the Invalid Mapping Data Detected dialog box appears, listing all the specific errors that were detected. You can choose to delete the invalid mappings and continue with only the valid mappings, or you can choose to keep the invalid mappings. However, notifications about invalid mappings will continue to appear as you continue to work with the integration. The system detects errors such as:

- Missing source or target fields in corresponding Robust Transform Engine (RTE) field mappings records
- Missing table columns in an import set

Note: In this situation, any corresponding metadata records in RTE are no longer valid and are automatically deleted. Records such as field mappings and transform operations that are associated with the missing table columns in the import set, are deleted.

- Missing an Identification and Reconciliation Engine (IRE) lookup rule for a lookup class

3. On the ETL Transform Map Assistant page, in the Specify Basic Details section of the guided setup, select the **Import Source Data and Provide Basic Details** task.
4. Fill out the form.

Field	Description
CMDB Application	The CMDB application associated with the ETL transform map. You can select Add new , which adds the CMDB Application and the Discovery Source fields for the new CMDB application.
Name	Name of the ETL transform map.
Description	Description of the integration.
Data Source	List of all data sources in the system.

Field	Description
	<p>Note: Be cautious in subsequently modifying the data source as it can result in substantial changes to the data integration. Aligning to the import set table of the new data source might require the removal of columns and associated transforms, or the addition of new columns. IntegrationHub ETL validation processes will detect any required updates and let you agree or reject these updates.</p>
Sample Import Set	<p>An Import Set that is associated with the specified Data Source.</p> <p>A subset of that Import Set data is used to preview source data.</p> <p>Select the Auto-pull a new import set option to pull a new Import Set of the associated data source.</p> <p>Starting with IntegrationHub ETL v3.2, if no Import Set is specified, then the map is loaded and is automatically set to be in read-only mode. You can review configurations in the map but can't edit mappings or transforms.</p>

Field	Description
Preview Size Override	<p>Number of data records that are loaded and used as sample for the preview for this transform map. If set, this custom setting overrides the value of the <code>sn_int_studio.preview.size</code> system property, and applies only to the current transform map.</p> <p>If Load Complete Schema is disabled, then the nested data structure for the map is generated based only on the specified number of records that are loaded.</p> <p>Field available starting with IntegrationHub ETL v3.2.</p>
Load Complete Schema	<p>Enable or disable loading the entire data schema for generating the data structure for the map.</p> <p>When disabled, the nested data structure for the map is generated based only on the number of records loaded as sample records for preview. The number of records loaded is determined by either the Preview Size Override setting, or by the global system property <code>sn_int_studio.preview.size</code>.</p> <p>Field available starting with IntegrationHub ETL v3.2.</p>

Field	Description
CMDB Application	Name of a new CMDB Application . Appears if you set CMDB Application to Add new .
Discovery Source	Discovery source associated with a new CMDB Application . Appears if you set CMDB Application to Add new .

5. Click **Save** to save the current changes or **Mark as Complete**.

A time stamp appears in the header when you click **Save**, which remains for the duration of the IntegrationHub ETL session for the ETL transform map. When you re-enter the session or switch between ETL maps, the time stamp disappears.

Preview and prepare data

Review sample records of raw source data, which will be integrated into the CMDB. Transform and prepare data to align with the target classes and attributes, if needed.

Before you begin

The number of records in the sample data is globally determined by the system property `sn_int_studio.preview.size`, which is set to 100 by default. The maximum number of records in the sample data that IntegrationHub ETL can process is 10,000. If you set that property above the 10,000 limit, then IntegrationHub ETL will only process up to 10,000 records and a message will appear to that effect.

Starting with IntegrationHub ETL v3.2, you can override the value of the `sn_int_studio.preview.size` property by setting the **Preview Size Override** field on the Import Source Data and Provide Basic Details form, per map.

To process nested data from a nested payload, the respective data source must be set with the **Data in single column** option.

Role required: cmdb_inst_admin

About this task

Review the values in the data columns of the sample data and identify columns that do not align with the requirements of the intended target classes and attributes. You can transform data, for example, by converting the data format, replacing values, and concatenating data columns. You can apply transformations one on top of another, creating a chain of data transformations. You can also set a data column to be ignored in the mapping and integration process.

Note: To set a CMDB attribute to be empty, use the string '<EMPTY_STRING>'.

Columns for nested data appear alongside the rest of the data, with a **Nested Objects** notation in the data column header. The count of nested data items per object appear with a link which lets you drill to deeper levels of the nested data. To show the data structure of nested data in a separate panel, enable the **Show data structure** option.

The Data Structure panel has two options for displaying nested data:

- **Tree:** Nested data grouped by objects, where each object node corresponds to a record entry in the source data. Expand object nodes to show all nested data for the record.
- **Collection:** Nested data grouped by the top-level object (by default) and then by nested data items such as software. Expand a node such as software, to show which software is installed on each computer.

You can navigate through the levels of nested data in the Data Structure panel, the breadcrumbs path, or through number links that appear in the source data itself. Your selections and the data that appears are kept synchronized between all views of the nested data, regardless of navigation.

For a demo about working with nested payload data, watch the [Integration Hub - ETL nested payload feature demo](#) video on the ServiceNow YouTube channel.

Procedure

1. Navigate to **All > Configuration > IntegrationHub ETL**, and click the **Name** of an integration.

The landing page of the IntegrationHub ETL lists all integrations that exist in the system, including integrations that were downloaded from the ServiceNow Store.

2. In the ETL Transform Map Assistant page, in the Prepare Source Data for Mapping section of the guided setup, select **Preview and Prepare Data**.
3. (Optional) Select **Show data structure** to open the Data Structure panel which shows the structure of nested data. In the Data Structure panel, you can drill down through the levels of nested data.
4. (Optional) Select the action menu for a column and then select a Sort operation.
5. Select the action menu for a column and then select **Group by** to group the data by the respective column. Select **Ungroup** to undo the grouping operation.
6. (Optional) Click **New Transform** and then select **Use Source Column**. Or, select the action menu for a column, and then select **New Transform** to transform the selected column.

You can't create new transforms for nested objects at this top-level view of the data. A nested object column contains number links which indicate the number of nested items for the record. To create a new transform for nested objects, click that number link to drill down to the actual nested data. Alternatively, navigate in the Data Structure panel to the nested object for which you want to create a transform.

A transform of nested data can reference parent objects of the nested data being transformed. Using the [sample payload for nested data](#) as an example, a transform for an interface object can reference the parent computer object but can't reference a software object.

- a. In the New Transform sidebar on the right, select a **Transform Type** and modify the **Transform Description** if appropriate. For more details about transform types, see [Transform types in IntegrationHub ETL](#).
- b. (Optional) Select **Hide initial column used for this transform** to hide from the current view all the columns that were used for this transform.

This setting is temporary for the current session, and if you refresh the page, the hidden column reappears. To show a hidden column, you can also click the gear icon on the banner frame. Then, move the hidden column from the Available to the Selected list and click **OK**.

- c. Select or verify the **Input Column** whose values are being transformed.
 - d. (Optional) Modify the **Output Column Name** for any of the columns that will be added with the transformed values.
 - e. Click **Apply**.
A new column with the transformed values appears, placed in alphabetical order based on the output column name. If you used the suggested output column name, then the new column appears to the right of the input column.
 - f. Review the transformed data and adjust any transforms, if needed.
7. (Optional) To apply the 'Set Fixed Value Column' transform:
- a. Click **New Transform** and then select **Set Fixed Value Column**.
 - b. In the Set Fixed Value Column sidebar, enter a **Column Name** and a **Column Description** for the new column. Then, set **Assign Column Value** to the value that is fixed for the new column.
 - c. Click **Apply**.
8. (Optional) Select the action menu for a column, and then select **Ignore in Mapping** to exclude the column from mapping and integration in the current session.
- In a subsequent session, the **Ignore in Mapping** setting does not apply, and the column will be included in mapping.
- You can click **Include in Mapping** to undo the **Ignore in Mapping** setting for the column.
9. (Optional) Select the action menu for a column, and then select **Delete This and Downstream Columns**. This delete action deletes the column along with any columns that were added using this column as an input column.

10. (Optional) Click **New Transform** and then select **Table Lookup** which lets you specify a table to look up and extract additional values from. Fill out the fields in the Table Lookup sidebar on the right. Values from the specified lookup table are matched with the mapped data. For the records that match, the specified values from the lookup table, are added as a column, to the data that is being prepared for mapping.

Table Lookup

Field	Description
Lookup Table	Table to use for matching with the data that is being mapped. When records from the lookup table and the mapped data satisfy the Look Up Condition , then specified values from the Lookup Table are extracted from the respective record and added to the mapped data.
Lookup Condition	A set of pairs of column conditions. Each pair specifies a column in the lookup table and a column in the mapped data, which are attempted to be matched. <ul style="list-style-type: none">• If values of target table column: The column in the target table to match to a column in the mapped data.• Match values of source data table: The column in the mapped data to match to a column in the lookup table. You can add multiple pairs of columns to match on.

Field	Description
Lookup Condition	<p>Values to extract from the Lookup Table when there is a match with the mapped data.</p> <p>Then output values from the following columns: The lookup table columns to extract values from, when values from the lookup table and the mapped data, satisfy the Lookup Condition.</p> <p>You can specify multiple lookup table columns to extract values from. For every column that you specify, a corresponding Output Column Name field automatically appears. Specify a label for the column that will be added with the extracted values.</p>
Output Column Name	<p>A label for the column that will be added to the mapped data, with the values extracted from the lookup table.</p> <p>An Output Column Name field is automatically added for every column that you specify in Then output values from the following columns.</p>

- Review the data and ensure that the intended set of data to be integrated is transformed, correctly formatted and prepared for import.

12. Click **Mark as Complete**.

Result

Data is prepared when the set of source data columns and transformed columns that you want to integrate, meet any formatting and other value requirements of the target CMDB classes and attributes. These columns are then ready to be mapped and integrated to CMDB classes and attributes.

About mapping data columns to CMDB classes and attributes

There are several requirements and guidelines for mapping source data to target CMDB classes and attributes. Also, there is an option of deactivating class mappings while preserving the settings for an easy reactivation. Review these concepts to ensure proper processing by the Identification and Reconciliation Engine (IRE).

Required mappings

You must map data to all required attributes of the target class in addition to mapping to attributes that are not configured as required. Also, the following two fields appear by default and you cannot delete them:

Source Native Key

IRE uses to uniquely identify a record and for building relationships and references. Also, improves performance of insert and update operations. When processing a payload, IRE generates an error if this field is empty.

Source Recency Timestamp

IRE uses to identify records that are older than the current record and therefore can be ignored, to help resolve conflicting attribute values. If a value is provided, it is used only if it is later than the value that is currently stored in the CMDB. If a value is not provided, IRE updates the attribute with the current timestamp.

The following system properties let you modify how IRE uses the source_recency_timestamp value in a payload to update the last_scan attribute in the Source [sys_object_source] table:

- `glide.identification_engine.skip_updating_last_scan_if_older`

- `glide.identification_engine.ire_message_listener_skip_updating_last_scan_to_now`

For more information about how IRE uses `source_native_key` and `source_recency_timestamp` for CI identification, see [Identification and Reconciliation engine \(IRE\)](#).

Conditional class

A conditional class lets you map different sets of data records to different target classes according to specific column values, or the status of a specific plugin.

For example, if a display name contains 'Windows', then 'Windows Server' is selected as the target class. But if the display name contains 'Linux', then 'Linux Server' is selected as the target class. For records that do not meet any of these conditions (display name does not contain 'Windows' nor 'Linux'), 'Server' is selected as the target class.

Associated class

An associated class lets you select the CMDB class to be associated with a target non-CMDB table. Setting an associated class is required for IRE processing if the non-CMDB table is not configured for IRE processing. For a non-CMDB table that is supported and configured for IRE processing, setting an associated class is optional. See [IRE support for non-CMDB tables](#) for more information.

The software Instance is a non-CMDB class but it does not have IRE rules associated with it. So, things we said about it here pre-Utah are still valid. But for non-CMDB classes with IRE rules it's not mandatory to have an association. For example "If the target class for mapping is a non-CMDB class with a reference to a CMDB class, you must select the CMDB class to associate the non-CMDB target class with" non-CMDB class with IRE rules Instead of "you must" it should be. "You can". Same with the Example it's not valid for non-CMDB with IRE rules.

If the target class for mapping is a non-CMDB class with a reference to a CMDB class, you must select the CMDB class to associate the non-CMDB target class with. A non-CMDB class refers to a class, such Serial Number [cmdb_serial_number], that does not extend the Configuration Item [cmdb_ci] class. The Related Entry [cmdb_related_entry] class might contain multiple CMDB class associations for the same non-CMDB class.

Therefore, select the appropriate association to allow IRE processes to update the target non-CMDB class.

For example, the Related Entry [cmdb_related_entry] class has a record which associates the non-CMDB Software Instance [cmdb_software_instance] class with the CMDB Software Package [cmdb_ci_spkg] class. If you select Software Instance as a target class, you must associate the Software Instance class with the Software Package [cmdb_ci_spkg] class.

Deactivating class mappings

When you edit an ETL transform map, provided by a Service Graph Connector for example, you can delete a class mapping to prevent the class from being populated when the integration runs. However, if you later decide to populate that class, you must readd that class and reconfigure all the class mappings. Instead, you can deactivate a class mapping to temporarily ignore the class during the integration run, while preserving all of its mapping configuration. A class that you choose to deactivate is grayed out in the user interface but you can continue and edit the class mappings. Later, you can reactivate a class mapping to enable populating the class, without needing to reconfigure the class mappings.

Some classes that you choose to deactivate, trigger an automatic deactivation of additional classes that you did not directly choose to deactivate. Which classes are automatically deactivated, depends on the class that you chose to deactivate. For example, whether the class has dependent relationships or associated classes. Those automatically deactivated classes:

- Appear in light gray in the user interface and you can't reactivate them.
- Are automatically reactivated when you reactivate:
 - The class that you initially deactivated which triggered the automatic deactivation
 - Any class that the deactivated class depends on

All classes that you directly deactivate mappings for and the resulting class mappings that are automatically deactivated, are not populated when the integration runs. Also, any relationships and lookup tables

associated with those classes, are not populated when the integration runs.

Class mapping and other deactivation scenarios:

- Deactivate a class which no class depends on and which has no associated classes:

Triggers an automatic deactivation of any lookup rules and relationships associated with the deactivated class.

- Deactivate a lookup rule, such as serial number, within a class mapping:

Does not trigger any automatic deactivations.

- Deactivate a CMDB class which is associated with a non-CMDB class:

• Triggers an automatic deactivation of the associated non-CMDB class.

• Deactivating the non-CMDB class, does not impact the associated CMDB class.

- Deactivate a class with dependent relationships (Applies only if the dependent relationship exists in IntegrationHub ETL):

• Triggers an automatic deactivation of any class that has a single dependent relationship with the deactivated class.

•

If a class has multiple dependent relationships, then it is automatically deactivated only when you deactivate all of the dependent on classes.

For example, a scenario in which the File System class has dependent relationships with both, the Computer and a Server class.

If you deactivate the Computer class, the File System class is not automatically deactivated. Only if you also deactivate the Server class, the File System class is automatically deactivated.

- Deactivate a conditional class or a class mapping within a conditional class:

- Deactivating or activating a conditional class, triggers an automatic deactivation or activation of all conditional class mappings within the conditional class.
-

Deactivating a class mapping within a conditional class: Prevents the deactivated class from getting populated during integration runs. However, the associated 'If', 'Else if', or 'Else' conditions themselves remain in effect within the condition of the conditional class. For example, if you deactivate the following class mapping:

[If] [operating_system] [contains] [Linux] Then [Class] [is] [Linux Server].

Then, the Linux Server class is not populated, but the **[If] [operating_system] [contains] [Linux]** condition is in effect.

Map data columns to CMDB classes and attributes

Choose target classes and attributes in the CMDB to map source data columns to. You can map a data column to a specific target class, or add conditions so that the choice of target class depends on specific data values.

Before you begin

Role required: cmdb_inst_admin

About this task

Data columns that you map can be either source data columns which were not transformed, or transformed data columns. For example, to integrate a data column into the Computer and Software Package classes, select those classes as target classes and then map data columns into specific attributes in those classes.

When you configure mapping for a class, relationship, or a lookup rule, those items are always initially set as activated. For details about the results of deactivating mappings, see [Deactivating class mappings](#).

Note: Changing a class impacts any mappings that were already configured for the class, sometimes deleting those mappings. Details about the affected mappings and the impact, appear in the Affected mappings dialog box before you proceed with the class change. However, these details appear only when the change is from a CMDB class to another CMDB class or from a non-CMDB class to another non-CMDB class.

Procedure

1. Navigate to **All > Configuration > IntegrationHub ETL**, and click the **Name** of an integration.
The landing page of the IntegrationHub ETL lists all integrations that exist in the system, including integrations that were downloaded from the ServiceNow Store.
2. In the ETL Transform Map Assistant page, in the Map Data to CMDB and Add Relationships section of the guided setup, select **Select CMDB Classes to Map Source Data**.
Attributes that are configured as required in the platform, are noted, and you must map a data column to each of those attributes.
3. Click **Add Class** to add a target class to map to, or click **Edit Class** to edit a class.
 - a. In the Add Class dialog box, select a CMDB **Class**.
 - b. Click **Save**.
 - c. (Optional) Set the **Activate/Deactivate Mapping** toggle switch for a class, to on or off. If the Affected class mappings dialog box appears, review the list of affected classes, and then click **Proceed**.
When you add a non-CMDB class, it is initially deactivated and the **Activate/Deactivate Mapping** toggle switch is disabled, until you add an associated class that is active.
4. Click **Add Conditional Class** and then in the **Add Conditional Class** dialog box, specify the conditions that must be met for data to be mapped to different target classes.
 - a. **Collection** is automatically set to the data branch in the hierarchy which is associated with the lowest-level attribute. You

can modify the value to the data branch from which you want to map data from, which must be at a higher level in the same data branch of the hierarchy.

- b. In the **If** drop-down list, select attribute conditions that data values must meet, or enter plugins in the search box and specify a plugin condition. You can then specify that the rest of the records, which did not match any conditions, are mapped to yet a different target class. Data records will be mapped to different target classes according to the conditions met.

When processing nested data, a prefix denotes the first level in the nested hierarchy for attribute items.

Note: When you select a non-CMDB class, it is initially deactivated and the **Activate/Deactivate Mapping** toggle switch is disabled, until you add an associated class that is active.

- c. Click **Save**.
- d. (Optional) Set the **Activate/Deactivate Mapping** toggle switch for a conditional class, to on or off. If the Affected class mappings dialog box appears, review the list of affected classes, and then click **Proceed**.
- e. (Optional) Click **Edit Class** to edit the settings of a conditional class. In the Edit Conditional Class dialog box, set the **Activate/Deactivate Mapping** toggle switch for a class mapping, to on or off. Click **Save**, and if the Affected class mappings dialog box appears, review the list of affected classes and then click **Proceed**.
 - A deactivated class is not populated during integration runs, however, this doesn't affect the associated condition. The 'If', 'Else if', and 'Else' conditions themselves remain in effect within the condition of the conditional class and matching Cls are filtered accordingly.
 - The toggle switch of the conditional class reflects the summary of the states of all the conditional class mappings within the conditional class. If at least one of the conditional class mappings is activated, then the toggle switch of

the conditional class appears as activated. Otherwise, the toggle switch of the conditional class appears as deactivated.

5. For a non-CMDB class, click **Add Associated Class** to associate the non-CMDB class with a CMDB class and to enable the **Activate/Deactivate Mapping** toggle switch.. Or, click **Edit Associated Class** to edit an already associated class.
 - a. In the Add Associated Class dialog box, select a CMDB class. The list includes all entries in the Related Entry [cmdb_related_entry] class for the specified non-CMDB table (deactivated classes are not included).
 - b. Click **Add**.
 - c. (Optional) Set the **Activate/Deactivate Mapping** toggle switch for an associated class, to on or off.
6. Click **Set Up Mapping** to configure mapping for a newly added class, or click **Edit Mapping** to edit a mapping.
 - a. To map, drag data columns from the Data sidebar on the right, to CMDB target attribute on the left side of the mapping page.



Or, click the icon to search and select data columns for the mapping.

When mapping nested data:

- Data columns in the Data sidebar appear in a tree format that represents the structure of the nested data. Each attribute is associated with sample data for the attribute.
- Transformed columns are noted by a cyan-shaded dot.
-

All mappings to a specific CMDB class must be from the same source branch in the nested data. Only the branch

from which you selected the first column to map, is valid for selecting columns in subsequent mappings.

This restriction applies differently when mapping to attributes in lookup tables. All mappings to attributes in a lookup table also must be from the same source branch. However, that source branch can be different than the source branch you used with non-lookup tables.

Note: You can work around this restriction by using the [Copy](#) transform in the data preparation step, to copy attributes from a parent level to a child level. Prepare the data so that all the attributes that you want to map, are at the same level.

- When you drag a column to map from the Data sidebar, the fields of CMDB target attributes that are valid for the mapping, are highlighted by a green frame. If you attempt to drop a column in an invalid target attribute, the respective field is highlighted by a red frame and an error appears.
- b. Click **Add Attribute**. Then, in the Add Attribute dialog box, from the **Attribute** list, select one or more items as target attributes to map data to. You can also scroll down to the **IRE Settings** section of the list and select one of the [robust import set transformer properties](#). Click **Save**.
For information about precedence order between robust import set transformer properties defined at the individual item level and at the IRE payload level, see [robust import set transformer properties](#).
- c. Map any lookup rules such as the 'Serial Number Lookup 1' rule.

Lookup rules are in a deactivated state until you map them. Click the filter icon for the lookup rule to edit or add any lookup filters. In the lookup filter dialog box, specify attribute or plugin conditions that must be met for data to be mapped to various target classes. Then click **Save**.

After mapping a field of a lookup rule, you can set the Activate/Deactivate Lookup rule toggle switch for a rule, to on or off.

- d. (Optional) Click **View Class Details** to view the current class in [CI Class Manager](#).
 - e. (Optional) Click the **Transform Data** tab to navigate to the data preparation page where you can review and further transform data that you want to map.
 - f. Return to the Select CMDB Classes to Map Source Data page.
7. Click **Mark as Complete**.

Add Relationships

Add relationships that exist among the target CMDB classes, for an integration.

Before you begin

- A class that you want to add in the relationship, must be in an activated state.
- A base relationship or a relationship within a conditional relationship, that you want to edit, must be in an activated state.
- In a conditional relationship that you want to edit, at least one relationship condition must be in an activated state. Otherwise, the **Edit Relationship** button is grayed out.

Role required: cmdb_inst_admin

About this task

When creating relationships with nested data, you can't create a relationship between sibling objects from the nested data. Using the [sample payload for nested data](#) as an example, you can't create a relationship between interfaces and software.

ITOM Visibility, if available, uses enhanced discovery patterns to identify and add CI relationships to the Suggested Relationships table in the base system. When applicable, use the Suggested Relationships table to select relationships that are in compliance with Common Service Data Model (CSDM) standards.

Procedure

1. Navigate to **All > Configuration > IntegrationHub ETL**, and click the **Name** of an integration.
The landing page of the IntegrationHub ETL lists all integrations that exist in the system, including integrations that were downloaded from the ServiceNow Store.
2. In the ETL Transform Map Assistant page, in the Map Data to CMDB and Add Relationships section of the guided setup, select **Add Relationships**.
3. To add relationships, select **Add Relationship** or **Add Conditional Relationship** if you want to specify attribute conditions that must be met before adding a relationship. Then, complete the following actions as needed.

Option	Description
Add Relationship	<ol style="list-style-type: none">a. Select the Parent, Child, and Relationship Type values.b. Click Add.
Add Conditional Relationship	<ol style="list-style-type: none">a. In the choose field list, select attribute conditions that the data values must meet.b. Select the Parent, Child, and Relationship Type values.c. Click Save. <p>When processing nested data, a prefix denotes the first level in the nested hierarchy for attribute items.</p>

The **Relationship Type** list menu changes based on the selected parent and child class:

- If there is a dependent relationship, the list is disabled and the relationship type is automatically populated.

- If there is more than one dependent relationship, the list displays both containment and hosting relationship options and the containment relationship type is automatically populated.
 - If there is no dependent relationship, the list displays **Suggested relationships** with the first suggested relationship automatically selected, followed by the base system relationship types.
 - If there is no suggested relationship, the list displays **No suggested relationships** followed by the base system relationship types.
4. Click **Save** to save the current changes or **Mark as Complete**.

A time stamp appears in the header when you click **Save**, which remains for the duration of the Integration Hub ETL session for the ETL transform map. When you re-enter the session or switch between ETL maps, the time stamp disappears.

Preview mapping results

Preview the results of the sample data integration.

Before you begin

Role required: cmdb_inst_admin

About this task

Run an integration test and view a summary of the results, for the sample data (by default, up to 100 records). The summary includes total numbers for relationships that were created, mapped classes, partial and incomplete payloads that IRE couldn't process. You can also view detailed messages from Robust Transform Engine (RTE) and from Identification Reconciliation Engine (IRE).

Note: Most IntegrationHub ETL log messages (from RTE and IRE) are informational. However, even if the com.glide.import_set.importlog_level and the glide.importlog.log_to_table system properties are set to not add INFO log messages, IntegrationHub ETL does render INFO log messages. For more details about these properties, see [Import sets properties](#).

After you view the details in the summary page, you can return to any step to make adjustments and then rerun the integration.

Procedure

1. Navigate to **All > Configuration > IntegrationHub ETL**, and click the **Name** of an integration.
The landing page of the IntegrationHub ETL lists all integrations that exist in the system, including integrations that were downloaded from the ServiceNow Store.
2. In the ETL Transform Map Assistant page, in the Preview Sample Integration Results and Schedule Import section of the guided setup, select **Test and Rollback Integration Results**.
3. On the Test and Rollback Integration Results page, click **Run Integration**.
4. View the summary page and click the various tabs to see the integration run results for the affected CMDB classes. You can click  to open CI forms and view information.

Note: The order of the attribute columns follows the default columns list for the class in the platform. First, the default columns for the class appear from left to right, followed by the rest of the attribute columns organized in alphabetical order. For example, to see the default columns list for the Computers class, navigate to **All > Configuration > Computers**.
5. (Optional) Select any class tab and click **Edit Mapping** to return to the Select CMDB Classes to Map Source Data page where you can review and change mapping settings.

Note: Clicking **Edit Mapping** rolls back all the changes that were made to the CMDB as a result of this integration run.
6. (Optional) Click the **Relationships** tab and review any relationships that were created. Click **Edit Relationships** to return to the Add Relationships page where you can review and change any relationship configurations.

Note: Clicking **Edit Relationships** rolls back all the changes that were made to the CMDB as a result of this integration run.

7. Click the **Error Log**, **Activity Log**, or the **Warning Log** tabs to see the respective details logged by IRE and RTE during the integration.

IRE log records are grouped by categories and further organized by the respective class. For IRE log messages, the Message column contains only the messages themselves which were extracted from the raw log message. The Log Message column contains the complete log message, which includes class and category in addition to the message itself. RTE logs appear under the Other category.

Use the **Verbose** toggle switch to change the viewing mode for the Message and the Log Message columns:

- Verbose on: Shows fully expanded text of log messages.
- Verbose off: Shows a condensed version of the log messages. The fully expanded text of the log messages appears when you point to a message.

8. Click the **Incomplete Payloads** and **Partial Payloads** tabs for details about IRE payloads for the integration run.
9. Select **Mark as Complete**. The Rollback options dialog box appears and you can choose either of the following options.
 - **Retain Data**: All the changes to the CMDB resulting from this integration, are retained.
 - **Perform Rollback**: All the changes to the CMDB resulting from this integration, are rolled back and the CMDB is restored to its state before running the integration.

Provide integration schedule

Configure a schedule for importing data to CMDB using this ETL Transform Map.

Before you begin

Role required: cmdb_inst_admin

Procedure

1. Navigate to **All > Configuration > IntegrationHub ETL**, and click the **Name** of an integration.
The landing page of the IntegrationHub ETL lists all integrations that exist in the system, including integrations that were downloaded from the ServiceNow Store.
2. In the ETL Transform Map Assistant page, in the Preview Sample Integration Results and Schedule Import section of the guided setup, select **Set Import Schedule**.
3. On the Provide Schedule page, click **Set Schedules**.
4. In the Scheduled Data Imports list view (which opens in a new tab), click **New**.
5. Fill out the Scheduled Data Import form and then click **Submit**.
See [Schedule a data import](#) for details about the form fields.
6. Click **Mark as Complete**.

Transform types in IntegrationHub ETL

Use various transforms in IntegrationHub ETL to convert and prepare source data for mapping to the CMDB.

Transforms from the [Integration Commons for CMDB](#) store app, are also available in IntegrationHub ETL.

Concatenation

Combines the values from input fields into a single string, joining them on the optional joining_string field.

Details	
Table	sys_rte_eb_concat_operation
Input fields	source_sys_rte_eb_fields
Output field	target_sys_rte_eb_field

Details	
Additional Fields	joining_string (optional)

Example	
Input	"input_1", "input_2", "input_3"
Additional Fields	joining_string = ", "
Result	"input_1, input_2, input_3"

Convert to Boolean

Converts the incoming value to a boolean. 'true' and '1' values convert to 'true' (case insensitive), and any other values convert to 'false'.

Details	
Table	sys_rte_eb_to_boolean_operation
Input fields	source_sys_rte_eb_field
Output field	target_sys_rte_eb_field

Examples:

- All of the following inputs return 'true':
 - true
 - 1
- All of the following inputs return 'false':
 - "input_1"
 - ""
 - 0

• 11

Convert to Date

Attempts to convert the incoming value to a GlideDateTime value by applying the date_format to the incoming value. Attempts to directly convert using GlideDateTime if the date_format is incorrect.

Details	
Table	sys_rte_eb_to_date_operation
Input fields	source_sys_rte_eb_field
Output field	target_sys_rte_eb_field Returns an empty value if unable to parse at all.
Additional Fields	date_format (Java simple date format)

Example	
Input	"2018/09/20 11:21:00 a.m. EST"
Additional Fields	date_format = "yyyy/MM/dd hh:mm:ss a z"
Result	"2018-09-20 16:21:00"

Example	
Input	"2018/09/20 01:21:00 PM EST"
Additional Fields	date_format = "yyyy/MM/dd hh:mm:ss a z"
Result	"2018-09-20 18:21:00"

Example	
Input	"09/20/18"
Additional Fields	date_format = "yyyy/MM/dd hh:mm:ss a z"
Result	"0018-09-20 00:00:00"

Convert to Numeric

Converts the incoming value to a number.

Details	
Table	sys_rte_eb_to_numeric_operation
Input fields	source_sys_rte_eb_field
	target_sys_rte_eb_field
Output field	If the incoming value is non-numeric, then the output is empty.

Example	
Input	1.23
Result	1.23

Example	
Input	1.00
Result	1

Example

Input	input_1
Result	null

Example

Input	two
Result	null

Copy

Copies the source field's value to all of the target fields.

Details

Table	sys_rte_eb_copy_operation
Input fields	source_sys_rte_eb_field
Output field	target_sys_rte_eb_fields
Additional Fields	overwrite_existing_value (optional, boolean): If true , then the values of target fields are replaced. Otherwise, any non-empty value is not overwritten.

Extract Leading Numeric

Sets the target field to be the first numeric value found in the source field.

Details

Table	sys_rte_eb_extract_numeric_operation
Input fields	source_sys_rte_eb_field

Details	
Output field	target_sys_rte_eb_field
Additional Fields	<ul style="list-style-type: none">decimal_places (optional, number): Forces the output to have a specified number of decimal places.remainder_target_field (optional, reference to a field): Set to the trimmed remainder of the source field, after removing the first numeric value.

Example	
Input	"100 mb"
Result	"100"

Example	
Input	"100.123 mb"
Result	"100.123"

Example	
Input	"100.123 mb"
Additional Fields	decimal_places = 2
Result	"100.12"

Example	
Input	"100 mb"

Example

Additional Fields	decimal_places = 2
Result	"100.00"

Example

Input	"100 mb"
Additional Fields	remainder_target_field = <field>
Result	"100" and <field> = "mb"

Glide Lookup

Performs a lookup in the database on the target_table.

Details

Table	sys_rte_eb.glide_lookup_operation
Input fields	source_sys_rte_eb_fields
Output field	target_sys_rte_eb_fields
Additional Fields	<ul style="list-style-type: none">• target_table• glide_matching_fields (string): Comma-separated list of column names in the target table. For each input field in source_sys_rte_eb_fields, there must be an equal number of values in glide_matching_fields• glide_target_fields (string): Comma-separated list of column names in the target table. For each target

Details	
	field in target_sys_rte_eb_fields, there must be an equal number of values in glide_target_fields.

Example	
Input	<ul style="list-style-type: none">Input Field 1: 100 South Charles Street, BaltimoreInput Field 2: MD
Additional Fields	<ul style="list-style-type: none">Target Table: Location (cmn_location)Glide Matching Fields: street,stateGlide Target Fields: sys_id
Result	Output Field 1: 25ab9c4d0a0a0bb300f7dabdc0ca7c1c

Min/Max

Sets the target field to either the maximum or minimum of the values from all input fields.

Details	
Table	sys_rte_eb_min_max_operation
Input fields	source_sys_rte_eb_fields
Output field	target_sys_rte_eb_field
Additional Fields	<ul style="list-style-type: none">data_type (choice list <STRING,NUMERIC,DATE>)

Details

- min_max (choice list <MIN,MAX>)

Example

Input	"2", "-1", "0"
Additional Fields	<ul style="list-style-type: none">• data_type = NUMERIC• min_max = MAX
Result	"2"

Example

Input	"a", "b"
Additional Fields	<ul style="list-style-type: none">• data_type = STRING• min_max = MAX
Result	"b"

Example

Input	"2", "-1", "0"
Additional Fields	<ul style="list-style-type: none">• data_type = NUMERIC• min_max = MIN
Result	"-1"

Example	
Input	“a”, “b”
Additional Fields	<ul style="list-style-type: none"> • data_type = STRING • min_max = MIN
Result	“a”

Multiple Input Script

Runs a script with multiple inputs, setting the target_field == output for that script.

Each source field is available inside of the ‘batch’ variable as JavaScript fields. The name of the JavaScript field is the field attribute of the entity field (looking at sys_rte_eb_field.field, not sys_rte_eb_field.name).

Details	
Table	sys_rte_eb_multi_in_script_operation
Input fields	source_sys_rte_eb_fields
Output field	target_sys_rte_eb_field
Additional Fields	<ul style="list-style-type: none"> • script (script) • use_unique_input_sets (boolean): When true, only unique input values are included in the data batch for IRE processing. Otherwise, all input object's field values are included.

Example for using use_unique_input_sets, with a script function that takes record_type and operating_system as input and returns record_with_os:

Input data

Record	record_type	operating_system	record_with_os
1	computer	Windows XP	
2	computer	Linux	
3	computer	Windows XP	

If `use_unique_inputs_sets` is set to `true`, then the script processes only two values (computer + Windows XP and computer + Linux). If `use_unique_inputs_sets` is set to `false`, then each of the three values is individually processed (computer + Windows XP, computer + Linux, and computer + Windows XP).

Sample script:

```
(function(batch, output) {
    for (var i = 0; i < batch.length; i++) {
        // batch[i] is the unique set of
        // inputs/individual record
        // batch[i].<field> gives access to
        // the field value
        var in0 = gs.nil(batch[i].record_
        type) ? '' : batch[i].record_type;
        var in1 = gs.nil(batch[i].operati_
        ng_system) ? '' : batch[i].operating_system;
        // output[i] is the output for the
        // specific combination of inputs/individual record
        output[i] = in0 + "_" + in1;
    }
})(batch, output);
```

Sample script:

```
/* Example Script
// In this example the script input fields are 'input_field_1', 'input_field_2' - replace these with the fields used as script inputs
// There is a static field 'input' that has all the input field values concatenated with a '|'
(function(batch, output) {
    for (var i = 0; i < batch.length; i++) {
        //step1: access the input variables
```

```
var a = batch[i].input_field_1; //Value of the first source field.  
var b = batch[i].input_field_2; //Value of the second source field.  
  
//step2: Your script/code goes here.  
var c = a + b;  
  
//step3: set the output for each elements  
output[i] = b;  
}  
  
})(batch, output);  
*/
```

Rexeg Replace

Replaces each substring of the incoming string that matches the specified match_regex, with the specified replacement_regex string value.

Details	
Table	sys_rte_eb_regex_replace_operation
Input fields	source_sys_rte_eb_field
Output field	target_sys_rte_eb_field
Additional Fields	<ul style="list-style-type: none">match_regex (string, regular expression)replacement_regex (string)

Example	
Input	"String&With(Special)\$Characters"

Example	
Additional Fields	<ul style="list-style-type: none">match_regex = “[^0-9a-zA-Z]+”replacement_regex = “ ”
Result	“String With Special Characters”

Replace

Replaces each substring in the incoming string that matches the specified match_string, with the replacement_string string value.

Details	
Table	sys_rte_eb_replace_operation
Input fields	source_sys_rte_eb_field
Output field	target_sys_rte_eb_field
Additional Fields	<ul style="list-style-type: none">match_string (string)replacement_string (string)

Example	
Input	“Original String”
Additional Fields	<ul style="list-style-type: none">match_string = “Original”replacement_string = “Replacement”
Result	“Replacement String”

Round Numeric

Rounds the number value to the nearest whole number. Non-numbers are truncated.

Details	
Table	sys_rte_eb_round_numeric_operation
Input fields	source_sys_rte_eb_field
Output field	target_sys_rte_eb_field

Example	
Input	"1.5"
Result	"2"

Example	
Input	"1.4"
Result	"1"

Example	
Input	"i'm a string"
Result	""

Script

Runs a script with input, setting the target_field == output for that script.

This transform has been superseded by the Multi Input Script transform and is included for backwards compatibility with existing configurations.

Details	
Table	sys_rte_eb_script_operation
Input fields	source_sys_rte_eb_field
Output field	target_sys_rte_eb_field
Additional Fields	<ul style="list-style-type: none"> • script (script) • use_unique_input_sets (boolean): When true, only unique input values are included in the data batch for IRE processing. Otherwise, all input object's field values are included. For an example and for more details, see the Multiple Input Script transform.

The source field is included in the 'batch' variable as the JavaScript field 'input'.

```
(function(batch, output) {
    for (var i = 0; i < batch.length; i++) {
        // batch[i] is the unique set of
        // inputs/individual record
        // batch[i].input gives access to
        // the field value
        var in0 = gs.nil(batch[i].input)
        ? '' : batch[i].input;
        // output[i] is the output for the
        // specific combination of inputs/individual record
        output[i] = in0 + " modified by script";
    }
})(batch, output);
```

Example:

```
/* Example Script
(function(batch, output) {
    for (var i = 0; i < batch.length; i++) {
        //step1: access the input variables
        var a = batch[i].input; //Value of the source file
```

ld.

```
//step2: Your script/code goes here.  
var b = a + 1;  
//step3: set the output for each elements  
output[i] = b;  
}  
})(batch, output);  
*/
```

Set

Sets the target field's value to the string specified in set_value.

Details	
Table	sys_rte_eb_set_operation
Input fields	source_sys_rte_eb_field
Output field	target_sys_rte_eb_field
Additional Fields	<ul style="list-style-type: none">set_value (string)overwrite_existing_value (optional, boolean): When true, the current value of the target field is overwritten. Otherwise, a non-empty value isn't replaced.

Split

Splits the source field's value on the splitting_string and assigns each resulting item from the split to the target_sys_rte_eb_fields, in order.

Details	
Table	sys_rte_eb_split_operation
Input fields	source_sys_rte_eb_field

Details	
Output field	target_sys_rte_eb_fields
Additional Fields	splitting_string (string)

Example	
Input	"value1 value2 value3", with target_sys_rte_eb_fields {target1,target2,target3}
Additional Fields	splitting_string = " "
Result	target1 : value1, target2 : value2, target3 : value3

Example	
Input	"value1 value2 value3", with target_sys_rte_eb_fields {target1}
Additional Fields	splitting_string = " "
Result	target1 : value1

Example	
Input	"value1", with target_sys_rte_eb_fields {target1,target2,target3}
Additional Fields	splitting_string = " "
Result	target1 : value1, target2 : <null>, target3 : <null>

Trim

Trims leading and trailing whitespace from the source_sys_rte_eb_field value and assigns the result to the target_sys_rte_eb_field. This transform is equivalent to a Java String.trim().

Details	
Table	sys_rte_eb_trim_operation
Input fields	source_sys_rte_eb_field
Output field	target_sys_rte_eb_field

Example	
Input	“ value 1 ”
Result	“value 1”

Uppercase

Uppercases the source_sys_rte_eb_field value and assigns the result to target_sys_rte_eb_field.

Details	
Table	sys_rte_eb_upper_case_operation
Input fields	source_sys_rte_eb_field
Output field	target_sys_rte_eb_field

Example	
Input	“value1”
Result	“VALUE1”

Uppercase Trim

Combines both the Uppercase and the Trim transforms.

Details	
Table	sys_rte_eb_upper_case_trim_operation
Input fields	source_sys_rte_eb_field
Output field	target_sys_rte_eb_field

Example	
Input	" value1 "
Result	"VALUE1"

Add custom before and after Java scripts for a data source of a CMDB integration application. Those scripts provide access to the input and output payloads of IRE. When a CMDB integration invokes Identification and Reconciliation Engine (IRE), those scripts run before and after IRE processes the integration payload.

Before you begin

Role required: cmdb_inst_admin

Procedure

1. Navigate to **All > Configuration > IntegrationHub ETL**.

The landing page of the IntegrationHub ETL lists all integrations that exist in the system, including integrations that were downloaded from the ServiceNow Store.

2. Click the **CMDB Application** link for the integration that you want to add scripts for.

3. On the CMDB Integration Studio Application form, in the CMDB Integration Studio Application Data Sources related list, open the data source record for the CMDB integration.
4. On the CMDB Integration Studio Application Data Source form, check **Execute Before Script** or **Execute After Script**. Review the **Before Script** and **After Script** comments and enter your custom scripts at the bottom of the script field.
The comments in the script fields provide details for the before and after scripts. Use those guidelines, explanations such as what operations are supported, and examples when creating your custom script.
5. Click **Update**.

Related concepts

- [Identification and Reconciliation engine \(IRE\)](#)

Use an existing ETL transform map if you need to create a map that is mostly similar to that existing map. Select an existing map, specify a new data source for the new duplicate map, and then change or retain other settings such as data transforms.

Before you begin

The data source for the new duplicated ETL transform map must satisfy these requirements:

- The schema of the data source must be identical to the schema of the original ETL transform map. For example, both data sources must have an identical number of columns and identical column labels.
- The data source must preexist.
- The data source must not be used in any other ETL transform map.

Role required: cmdb_inst_admin

Procedure

1. Navigate to **All > Configuration > IntegrationHub ETL**.

The landing page of the IntegrationHub ETL lists all integrations that exist in the system, including integrations that were downloaded from the ServiceNow Store.

2. Click **Duplicate** and then fill out the Duplicate ETL Transform Map form.

Field	Description
Duplicate from	The ETL transform map to duplicate from.
CMDB application	The CMDB application associated with the ETL transform map. You can select Add new , which adds the CMDB Application and the Discovery Source fields for the new CMDB application.
Discovery Source	Discovery source associated with a new CMDB Application. Appears if you set CMDB Application to Add new .
CMDB Application Name	Name of a new CMDB Application. Appears if you set CMDB Application to Add new .
Name	Name of the ETL transform map.
Data Source	The source feed, unique for this ETL transform map, where the raw source data is imported from.

3. Click **Create Duplicate**.

Result

Data for the new duplicated ETL transform map is imported, and the **Import Source Data and Provide Basic Details** task, is set as complete.

What to do next

Continue with the next steps on the guided setup steps to review the imported data and complete the integration using the duplicated ETL transform map.

Service Graph Connectors

Service Graph Connectors are the pre-defined integrations that ingest data into Configuration Management Database (CMDB) from third-party sources from different domains like security, server, software or monitoring, internet of things (IoT), and Cloud.

Important: See [Available Service Graph Connectors](#) for a list of connectors provided by ServiceNow.

Overview

As a purpose-built app, you can use a Service Graph Connector to maintain the quality and consistency of third-party data in your CMDB.

The following video provides an overview of Service Graph Connectors.

The connectors also make sure that third-party data is mapped to the right locations in your CMDB as specified by the Common Service Data Model (CSDM). The CSDM enables ServiceNow products to use the data and increases reporting accuracy. For more information on CSDM, see [Common Service Data Model](#).

The Service Graph Connectors manage the configuration data pipeline in the following steps:

1. Ingest the data by identifying class, attribute, and data sources by using the identification rules.
2. Standardize the data to comply with your CMDB.

3. Reconcile the data into a single coherent picture by using the reconciliation rules.
4. Ingest the data into your CMDB.

The guided setup provided within each connector walks you through the steps for configuration. You can then track the status and processing results of all installed integrations using the integrations dashboard, provided with the Integration Commons for CMDB store app. For more information, see [Integration Commons for CMDB \(2.11.1\)](#).

Note: The Integration Commons for CMDB store app provides a dashboard with a central view of the status, processing results, and processing errors of all installed integrations using Service Graph Connectors. For more information, see [CMDB Integrations Dashboard](#).

Key capabilities

Healthy data

Maintains data using the multisource CMDB.

Improved experience

Reduces customization and time to value by speeding up deployment times with intuitive setup options and guidance available through the guided setup provided with the Service Graph Connectors.

Reduced risk

Ensures that the Service Graph Connectors are supported and up-to-date with the certified partners.

Using Service Graph Connectors

You can use Service Graph Connectors for common applications from the ServiceNow Store.

By using Service Graph Connectors, you can mandate data governance and design practices. Some of the examples practices that are mandated using Service Graph Connectors are:

- Consistently associating data with specific CI types. For instance, IP addresses are always assigned to network interfaces rather than

a mix of interfaces and servers. Ensuring that ServiceNow products know where to find third-party data. For more information, see [CMDB classifications and class dependency](#).

- Using the Identification and Reconciliation (IRE) engine to identify and classify data correctly before it's loaded into the CMDB. Preventing duplicate CIs and ensuring that attribute values are consistent across multiple data sources.

For information on the IRE-related rules used in Service Graph Connectors, see the following topics:

- [CMDB Identification and Reconciliation](#)
- [Identification rules](#)
- [Reconciliation rules](#)
- [Create an IRE data source rule](#)
- [Detecting duplicate CIs](#)
- [Create a data refresh rule](#)
- [Create an identification inclusion rule](#)
- Using the IntegrationHub ETL functionality so that data is transformed and loaded in the fastest and most efficient manner. For more information, see [IntegrationHub ETL \(3.2\)](#).

Available Service Graph Connectors

You can use a Service Graph Connector to import and integrate third-party data into CMDB classes and properties.

Important: Visit the [ServiceNow Store](#) website to view the latest list of all Service Graph Connectors.

Select an application to learn about the Service Graph Connector available for ingesting data from the application into your CMDB.

Public cloud providers



Endpoint or IT asset management applications



Monitoring and observability applications



Software, server, or network applications



Operational Technology (OT) applications



Partner-built connectors

Service Graph Connectors can be built by partners and vendors. For a list of partner-built connectors, see the [ServiceNow Store](#) website.

Service Graph Connector for AWS (2.2.1)

Use the Service Graph Connector for AWS to securely bring in Amazon Web Services (AWS) data into your ServiceNow instance.

Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

The integration uses AWS native technologies and AWS security best practices to enable cloud teams to connect the data within their ServiceNow workflow. For more information about the Service Graph Connector for AWS, see the [Service Graph Connector for AWS - Introduction](#) article on the ServiceNow Community site.

Supported versions

Supported ServiceNow versions:

- Tokyo
- Utah
- Vancouver

Use Cases

The following are examples on how you can use the Service Graph Connector for different ServiceNow applications:

- Visibility into cloud resources, relationships, and state in real time.
- Deep discovery of Applications for ITAM/SAM outcomes.
- Governance and Compliance outcome.

Guided Setup

The guided setup for the Service Graph Connector for AWS provides an organized sequence of tasks to configure the integration on your instance. To access the guided setup, see [Configure guided setup](#).

CMDB Integrations Dashboard

The Integration Commons for CMDB store app provides a dashboard with a central view of the status, processing results, and processing errors of all installed integrations. You can see metrics for all integration runs. You can filter the view to a specific CMDB integration, a specific time duration, or a specific integration run. For more details about monitoring AWS integrations in the CMDB Integrations Dashboard, see [Using the CMDB Integrations Dashboard](#).

Data Mapping

Data from the AWS data sources is mapped and transformed into the ServiceNow CMDB Configuration Item (CI) class definitions using the Robust Transform Engine (RTE). Data is inserted into the ServiceNow CMDB using the Identification and Reconciliation Engine (IRE).

You can use the IntegrationHub ETL app to view the data maps. See [IntegrationHub ETL \(3.2\)](#) for more information.

Note: If the **Use last run datetime** field is empty, the connector will pull in the baseline data. If the field has a data, the connector will pull in new data.

The following tables lists the data sources included for AWS and the corresponding staging tables where the imported data is loaded.

Data sources and staging tables for AWS

Data source	Staging table
SG-AWS-API-Gateway	SG-AWS-API-Gateway [sn_aws_integ_sg_aws_api_gateway]
SG-AWS-Datacenters	SG-AWS-Datacenters [sn_aws_integ_sg_aws_datacenters]
SG-AWS-DynamoDb	SG-AWS-DynamoDb [sn_aws_integ_sg_aws_dynamodb]
SG-AWS-EC2	SG-AWS-EC2 [sn_aws_integ_sg_aws_ec2]
SG-AWS-EKS-Cluster	SG-AWS-EKS-Cluster [sn_aws_integ_sg_aws_eks_cluster]
SG-AWS-EKS-Cluster-2	SG-AWS-EKS-Cluster-2 [sn_aws_integ_sg_aws_eks_cluster_2]
SG-AWS-EKS-FULL	SG-AWS-EKS-FULL [sn_aws_integ_sg_aws_eks_full]
SG-AWS-ELB-V1	SG-AWS-ELB-V1 [sn_aws_integ_sg_aws_elb_v1]
SG-AWS-ELB-V2	SG-AWS-ELB-V2 [sn_aws_integ_sg_aws_elb_v2]

Data source	Staging table
SG-AWS-Hardware-Type	SG-AWS-Hardware-Type [sn_awx_integ_sg_aws_hardware_type]
SG-AWS-Image-Id	SG-AWS-Image-Id [sn_awx_integ_sg_aws_image_id]
SG-AWS-Image-Private	SG-AWS-Image [sn_awx_integ_sg_aws_image]
SG-AWS-Lambda	SG-AWS-Lambda [sn_awx_integ_sg_aws_lambda]
SG-AWS-Network-Interface	SG-AWS-Network-Interface [sn_awx_integ_sg_aws_network_interface]
SG-AWS-Organization	SG-AWS-Organization [sn_awx_integ_sg_aws_organization]
SG-AWS-RDS	SG-AWS-RDS [sn_awx_integ_sg_aws_rds]
SG-AWS-S3	SG-AWS-S3 [sn_awx_integ_sg_aws_s3]
SG-AWS-Security-Group	SG-AWS-Security-Group [sn_awx_integ_sg_aws_security_group]
SG-AWS-Service-Account	SG-AWS-Service-Account [sn_awx_integ_sg_aws_service_account]
SG-AWS-Software-Inventory	SG-AWS-Software-Inventory [sn_awx_integ_sg_aws_software_inventory]

Data source	Staging table
	SG-AWS-Software-Staging [sn_aws_integ_sg_aws_temp_software_staging]
SG-AWS-Software-Remove	SG-AWS-Software-Remove [sn_aws_integ_sg_aws_software_remove]
SG-AWS-SSM-SendCommand	SG-AWS-SSM-SendCommand [sn_aws_integ_sg_aws_ssm_sendcommand]
SG-AWS-Storage-Volume	SG-AWS-Storage-Volume [sn_aws_integ_sg_aws_storage_volume]
SG-AWS-Subnets	SG-AWS-Subnets [sn_aws_integ_sg_aws_subnets]
SG-AWS-Tags	SG-AWS-Tags [sn_aws_integ_sg_aws_tags]
SG-AWS-VM-Hw-Consolidation	SG-AWS-VM-Hw-Consolidation [sn_aws_integ_sg_aws_vm_hw_consolidation]
SG-AWS-VPC	SG-AWS-VPC [sn_aws_integ_sg_aws_vpc]

The imported data from staging tables is then inserted into the following target tables:

- Application [cmdb_ci_appl]
- Availability Zone [cmdb_ci_availability_zone]
- AWS Datacenter [cmdb_ci_aws_datacenter]
- Block Endpoint [cmdb_ci_endpoint_block]
- Cloud Database [cmdb_ci_cloud_database]

- Cloud Function [cmdb_ci_cloud_function]
- Cloud Gateway [cmdb_ci_cloud_gateway]
- Cloud Load Balancer [cmdb_ci_cloud_load_balancer]
- Cloud Mgmt Network Interface [cmdb_ci_nic]
- Cloud Network [cmdb_ci_network]
- Cloud Object Storage [cmdb_ci_cloud_object_storage]
- Cloud Organizations [cmdb_ci_cloud_org]
- Cloud Service Account [cmdb_ci_cloud_service_account]
- Cloud Subnet [cmdb_ci_cloud_subnet]
- Compute Security Group [cmdb_ci_compute_security_group]
- Docker Container [cmdb_ci_docker_container]
- Docker Image [cmdb_ci_docker_image]
- DynamoDB Table [cmdb_ci_dynamodb_table]
- Hardware Type [cmdb_ci_compute_template]
- Image [cmdb_ci_os_template]
- IP Address [cmdb_ci_ip_address]
- Key Value [cmdb_key_value]
- Kubernetes Cluster [cmdb_ci_kubernetes_cluster]
- Kubernetes DaemonSet [cmdb_ci_kubernetes_daemonset]
- Kubernetes Deployment [cmdb_ci_kubernetes_deployment]
- Kubernetes Namespace [cmdb_ci_kubernetes_namespace]
- Kubernetes Node [cmdb_ci_kubernetes_node]
- Kubernetes Pod [cmdb_ci_kubernetes_pod]
- Kubernetes Service [cmdb_ci_kubernetes_service]

- Kubernetes Volume [cmdb_ci_kubernetes_volume]
- Network Adapter [cmdb_ci_network_adapter]
- Running Process [cmdb_running_process]
- Server [cmdb_ci_server]
- Software Installation [cmdb_sam_sw_install] (If SAM is installed.)
- Software Instance [cmdb_software_instance] (If SAM is not installed.)
- Software [cmdb_ci_spkg] (If SAM is not installed.)
- Storage Mapping [cmdb_ci_storage_mapping]
- Storage Volume [cmdb_ci_storage_volume]
- Storage Volume Snapshot [cmdb_ci_storage_vol_snapshot]
- TCP [cmdb_tcp]
- Virtual Machine Instance [cmdb_ci_vm_instance]
- VNIC Endpoint [cmdb_ci_endpoint_vnic]

Note: If the AWS Systems Manager (SSM) service isn't enabled, the connector populates the server records in the Server [cmdb_ci_server] class. If the AWS SSM service is enabled, then based on the platform type obtained through the SSM service, the server records are populated in either the Linux Server [cmdb_ci_linux_server] class or the Windows Server [cmdb_ci_win_server] class. The Server [cmdb_ci_server] class is the parent class of the Linux Server [cmdb_ci_linux_server] and the Windows Server [cmdb_ci_win_server] classes.

For more information on where data is saved when pulling data from AWS, see [CMDB classes targeted](#).

When you run the diagnostic test, the data is loaded in the following tables:

- SG AWS Diagnostic Details [sn_aws_integ_sg_aws_diagnostic_details]

- SG-AWS Diagnostic Summary
[sn_aws_integ_sg_aws_diagnostic_summary]
- SG AWS Diagnostic Summary Notes
[sn_aws_integ_sg_aws_diagnostic_summary_notes]

The AWS configuration data for each connection is stored in the SG AWS Application properties [sn_aws_integ_sg_aws_application_properties] table.

For more information about how CI information is pulled from AWS, see the [Service Graph Connector for AWS - Functional Spec and CI](#) article on the ServiceNow Community site.

Additional resources

See the following articles on the ServiceNow Community site for any additional information on the AWS set up:

- [Service Graph Connector for AWS - Introduction](#)
- [Service Graph Connector for AWS - Diagnostic Tool & Troubleshooting Issues](#)
- [Service Graph Connector for AWS - Functional Spec and CI](#)
- [Cloud Discovery and SG-AWS](#)

Set up the AWS environment and scheduled jobs to pull in AWS data into the CMDB.

Before you begin

To use this Service Graph Connector, you need a subscription to a Subscription Unit that is based in the IT Operations Management (ITOM) Visibility application or in the ITOM Discovery application. As defined in the section titled "Managed IT Resource Types" in [ServiceNow Subscription Unit Overview](#), for managed IT resources that are created or modified in the CMDB by this Service Graph Connector, but that aren't yet managed by [ITOM Visibility or ITOM Discovery](#), these resources will increase Subscription Unit consumption from that application. Review your current Subscription Unit consumption within ITOM Visibility or ITOM Discovery to ensure available capacity.

Dependencies and requirements:

- The [Integration Commons for CMDB](#) store app, which is automatically installed.
- The [CMDB CI Class Models store app](#) store app, which is automatically installed.
- Discovery Core plugin (com.snc.discovery.core), which is automatically installed by Discovery.
- The ITOM Discovery License plugin (com.snc.itom.discovery.license). You must activate this plugin.
- ITOM Licensing plugin (com.snc.itom.license). For more information, see [Request Discovery](#).

Starting with the San Diego release, embedded help content won't be visible on the default Polaris theme in the Next Experience. You must activate the embedded help content by completing the following steps:

1. Select a task section in the guided setup, and then select **Configure**.
2. In the menu bar, select the help icon (ⓘ).

Role required: admin

About this task

For more information on the **Service Graph Connector for AWS** setup instructions, see the following articles:

- [Service Graph Connector for AWS - Setup Instructions \[KB1220597\]](#) article on the Now Support Knowledge Base.
- [Service Graph Connector for AWS - Introduction](#) on the ServiceNow Community site.
- [SGC-AWS - Release 2.0 Features](#) on the ServiceNow Community site.

Procedure

1. Ensure that you've selected the **Service Graph Connector for AWS** application scope by using the application picker.
For more information, see [Application picker](#).

2. Navigate to **All > Service Graph Connector for AWS > Setup**.
3. On the Getting started page, select **Get Started**.
4. Download the scripts.
 - a. In the Configure the AWS environment section of the Service Graph Connector for AWS page, select **Get Started**.
 - b. For the Download the scripts task, select **Configure**.
 - a. When you download all the scripts onto your local machine, select **Close** to close the Download the scripts dialog box and get return to the guided setup.
 - b. Set the Download the scripts task to complete by selecting **Mark as Complete**.
5. Create the required application properties for an AWS credential.
 - a. For the Add configuration properties for instance task, select **Configure**.
 - b. On the SG AWS Application properties form that opens in a new tab, fill in the fields.

SG AWS Application properties form

Field	Description
Connection Details	
Connection Alias	Name to identify the AWS connection record. For example, SG_AWS_CredentialAlias_0 rg. You can add multiple AWS instances. However, don't modify the name for the default connection alias SG_AWS_CredentialAlias_0 rg.

Field	Description
Organization Details	
Organization Account	Numeric account identifier of the AWS organization.
Organization Name	Name of the AWS organization.
Organization Description	Description of the AWS organization.
Standalone Account ID	ID of a member account in the AWS organization.
AWS Regions	
Regions	<p>AWS regions to collect the CI data.</p> <p>By default, the Service Graph Connector for AWS runs through all the AWS regions to collect the CI data.</p> <p>You can enter AWS specific regions to speed up the CI data import process. For example, <code>us-east1</code>, <code>us-east-2</code>.</p> <p>If this field is left empty, the Service Graph Connector for AWS pulls the resources from all the AWS regions.</p> <p>However, for the AWS GovCloud regions, don't leave the Regions field empty. The supported AWS GovCloud</p>

Field	Description
	<p>regions are us-gov-east-1 and us-gov-west-1.</p> <p>If you update the Regions field value later, clear the value of the Last run datetime field in all the Service Graph Connector for AWS-related data sources to import a new set of data.</p>
STS Assume Role Name	
STS Role	<p>AWS Identity and Access Management (IAM) role name that is obtained by the ServiceNow user by calling the AssumeRole API offered by the AWS Security Token Service (STS). The AssumeRole API returns a set of temporary security credentials for the ServiceNow user to access the AWS resources.</p> <p>Note: Enter the IAM role name but don't prefix <code>arn</code> in the name. If you leave this field is empty, the value of this field is automatically set to SnowOrganizationAccountAccessRole, which is the default IAM role name for the ServiceNow user.</p>
S3 Account Details	
S3 Account Id	Numeric identifier of the AWS account that hosts

Field	Description
	the Amazon Simple Storage Service (Amazon S3) bucket.
S3 Bucket Name	Name of the Amazon S3 bucket that collects the details from Amazon EC2 instances.
S3 Region	Region where the Amazon S3 bucket resides.
SSM SendCommand Document details	
SSM Send Command Linux Name	Name of the document that defines the actions run by the AWS Systems Manager (SSM) on a Linux-based Amazon EC2 instance.
SSM Send Command Windows Name	Name of the document that defines the actions run by the AWS SSM on a Windows-based Amazon EC2 instance.
Management Account ID	
Management Account ID	<p>Management account in the AWS organization. The account calls the <code>ListAccounts</code> API associated with the AWS organization to collect CI information from all the accounts. For more information, see ListAccounts on the AWS documentation site.</p> <p>Enter a value for this field when the ServiceNow user was created in an AWS member account.</p>

Field	Description
AWS Config Aggregator details	
Config Aggregator Account	<p>AWS account where the aggregator resource type in the AWS Config service has been configured.</p> <p>Enter a value in this field when you're using an AWS Config aggregator.</p>
Config Aggregator Name	Name of the aggregator resource type. This field is available only when you enter a value in the Config Aggregator Account field.
Config Aggregator Region	Region where the aggregator resource type resides. This field is available only when you enter a value in the Config Aggregator Account field.
AWS Key Rotation Setup	
AWS Rotate Keys	Option to enable the key rotation process.
AWS Key Rotation Date	Key rotation date. This field is automatically set to the next rotation date. This field is available only when you select the AWS Rotate Keys check box.
AWS Key Rotation Period (in Days)	Key rotation period in days. This field is available only when you select the AWS Rotate Keys check box.

Field	Description
AWS Key Rotation Status	Status message of a key rotation displaying whether the rotation was a success or a failure. This field is automatically set to display the key rotation status message. This field is available only when you select the AWS Rotate Keys check box. If the rotation status is a failure, an email notification is triggered, if configured.
Gov Cloud Setup	
Is Gov Cloud	Option to indicate that the connection setup is for the AWS GovCloud.

- c. Select **Update** and close the tab and return to the guided setup tab.
- d. Set the Add configuration properties for instance task to complete by selecting **Mark as Complete**.
6. (Optional) Configure the notification settings to receive email notifications on the AWS key rotation status.
 - a. For the Email Notification Setup task, select **Configure**.
 - b. On the Notification form that opens in a new tab, fill in the fields. For more information, see [Create an email notification](#).
 - c. Select **Update** and close the tab and return to the guided setup tab.
 - d. Set the Email Notification Setup task to complete by selecting **Mark as Complete**.
7. Configure the authentication credentials to authenticate requests sent to the AWS APIs.

- a. Configure your AWS credentials.
 - a. For the Configure the connection section of the Service Graph Connector for AWS page, select **Get Started**.
 - b. For the Configure the credentials task, select **Configure**.
 - c. In the **Name** field, enter a name for the authentication.
SG-AWS-Credentials-0rg is the default credential alias name. You can add multiple AWS instances. However, don't modify the default connection alias.
 - d. Enter the access key ID and the secret access key in the **Access Key ID** and Secret Access Key fields respectively.
The AWS access keys are long-term credentials for the IAM user and include two parts: an access key ID and a secret access key. You must use both the access key ID and the secret access key together to authenticate requests.
 - e. Return to the Configure the connection task page by selecting the back icon (<).
 - f. Set the Configure the credentials task to complete by selecting **Mark as Complete**.
- b. Test the AWS API connection to import data from the AWS application.
 - a. For the Test the connection task in the Configure the connection section, select **Configure**.
 - b. Select the **Test Load 20 Records** related link.
The Test Connection dialog box opens displaying the import progress.
 - c. When the progress state changes to **Complete**, select the back icon (<) to return to the guided setup.
 - d. Set the Test the connection task to complete by selecting **Mark as Complete**.

8. Configure the required EC2 resources for Amazon Elastic Kubernetes Service (EKS) to import EKS cluster data.

An EKS EC2 resource is a bastion host that has network access to EKS clusters. The EKS clusters aren't directly accessible to the connector. Therefore, you must provide the EKS EC2 resource details. For importing EKS cluster data, the connector uses the SSM Send Command on EKS EC2 resources to run kubectl commands remotely.

Note: Ensure that you've configured your AWS environment for the EKS integration. For more information, see the [Service Graph Connector for AWS - Amazon EKS Integration \[KB1437138\]](#) article in the Now Support Knowledge Base.

- a. For the Configure the EKS Resource Details section of the Service Graph Connector for AWS page, select **Get Started**.
- b. For the Enter the EKS EC2 Resource Details task, select **Configure**.
- c. On the form that opens in a new tab, fill in the fields.

SG-AWS-EKS-EC2-Resource form

Field	Description
Active	Option to activate the EKS EC2 resource. Note: Set to false, if you are not using the EKS EC2 resource resource.
EC2 Region	AWS region where the EKS EC2 resource is located.
EKS EC2 Resource Id	Identifier of the EKS EC2 resource.
EC2 Account	User name assigned to the EKS EC2 resource account.
Connection Alias	Connection alias associated with the AWS environment

Field	Description
	setup and configured in step 7.a.

- d. Select **Submit** to return to the guided setup.
 - e. Repeat steps from **8.b** to **8.d** to add multiple EKS EC2 resources.
All the EKS EC2 resources are added to the SG-AWS-EKS-Master [sn_aws_integ_sg_aws_eks_master] table.
 - f. Set the Enter the EKS EC2 Resource Details task to complete by selecting **Mark as Complete**.
9. Run the AWS diagnostic tool before running a scheduled import job to identify any issues in the AWS environment setup.
- a. For the Service Graph AWS Diagnostic Tool section of the Service Graph Connector for AWS page, select **Get Started**.
 - b. For the AWS Setup Diagnostic Tool task, select **Configure**.
 - c. Select the organization ID from the text field.
 - d. Select **Run Diagnostic Test**.

Tip: Select one of the following options to exclude the corresponding test results from the diagnostic summary:

Skip SSM setup tests

Excludes the software inventory data from the summary results by not calling the GetInventory API. Select this option when you've opted out or not set up the configuration for SSM.

Skip SSM Deep Discovery tests

Excludes the deep discovery data from the summary results. Select this option when you've opted out or not set up the configuration for SSM deep discovery.

Skip EKS setup tests

Excludes the EKS data from the summary results by not running the kubectl commands. Select this option when you've opted out or not set up the EKS integration.

- e. (Optional) View only EKS cluster test results by selecting **View EKS Test Details**.
 - f. (Optional) Preview any previous diagnostic tool results by selecting **Load DT Results**, selecting a diagnostic ID, and then selecting **Load Results**.
 - g. When you finish reviewing the diagnostic summary results, select the back button of your browser to return to the guided setup.
 - h. Set the AWS Setup Diagnostic Tool task to complete by selecting **Mark as Complete**.
10. Configure the scheduled jobs to import data from the AWS application.
- a. In the Configure the scheduled import jobs section of the Service Graph Connector for AWS page, select **Get started**.
 - b. For the Configure the scheduled job task, select **Configure**.
 - c. Select the scheduled job that you want to activate.

- d. On the Scheduled Data Import form, verify the field values for the scheduled job.

For more information, see [Schedule a data import](#).

- e. Select **Execute Now**.

- f. Repeat the steps [10.c](#) to [10.e](#) for each scheduled job for data import.

- g. Select the back icon (<) to return to the guided setup page.

- h. Set the Configure the scheduled job task to complete by selecting **Mark as Complete** in the guided setup.

11. (Optional) Add multiple AWS instances.

- a. In the Add Multiple Instances section of the Service Graph Connector for AWS page, select **Get Started**.

- b. Create a credential alias for the new AWS connection in the Service Graph Connector for AWS.

- a. For the Create new Connection & Credentials Alias Record task, select **Configure**.

- b. On the Connection & Credential Aliases form that opens in a new tab, fill in the connection details.

- c. Select **Submit** and close the tab and return to the guided setup tab.

- d. Set the Create new Connection & Credentials Alias Record task to complete by selecting **Mark as Complete**.

- c. Create credentials for the new AWS credential alias.

- a. For the Create new AWS Credentials task, select **Configure**.

- b. On the AWS Credentials form that opens in a new tab, fill in the credential name, access key, and secret key details.

- c. In the Credential alias field, select the unlock credential alias

icon () to select the credential alias you created in step [11.b](#).

- d. Set the Create new AWS Credentials task to complete by selecting **Mark as Complete**.
- d. Create application properties for the new AWS connection by selecting **Configure** for the Configure AWS environment for the new Instance task.
For more information, follow the step 5 discussed earlier for configuring the application properties of the AWS connection available by default.
- e. Create data sources for the new AWS connection.
 - a. Ensure that you have edit permissions for the Datasource [sys_data_source] table.
 - b. For the Update Data Source Access task, select **Configure**.
 - c. To edit the record, select the **Global** application scope from the application picker.
 - d. In the Application Access related list of the Data Source form that opens in a new tab, select the **Can create**, **Can update**, and **Can delete** check boxes.
 - e. Select **Update**.
 - f. Select the back icon (<) to return to the guided setup page tab.
 - g. From the application picker, select the **Service Graph Connector for AWS** application scope.
 - h. Set the Update Data Source Access task to complete by selecting **Mark as Complete**.
- f. Create a scheduled import job for the new AWS connection.
 - a. Ensure that you have edit permissions for the Scheduled data import [scheduled_import_set] table.
 - b. For the Update Scheduled Data Import Access task, select **Configure**.
 - c. To edit the record, select the **Global** application scope from the application picker.

- d. In the Application Access related list of the Scheduled Data Import form that opens in a new tab, select the **Can create**, **Can update**, and **Can delete** check boxes.
- e. Select **Update**.
- f. Select the back icon (<) to return to the guided setup page tab.
- g. From the application picker, select the **Service Graph Connector for AWS** application scope.
- h. Set the Update Scheduled Data Import Access task to complete by selecting **Mark as Complete**.
- g. Clear the cache on the Data Source [sys_data_source] and Scheduled Data Imports [scheduled_import_set] tables.
 - a. For the Clear Cache for Data Source and Scheduled Data Imports tables task, select **Configure**.
 - b. To edit the record, select the **Global** application scope from the application picker.
 - c. In the **Run script** field, enter the following code:

```
GlideTableManager.invalidateTable("sys_data_source");
GlideCacheManager.flushTable("sys_data_source");
GlideTableManager.invalidateTable("scheduled_import_set");
GlideCacheManager.flushTable("scheduled_import_set");
GlideTableManager.invalidateTable("sys_db_object");
GlideCacheManager.flushTable("sys_db_object");
```

- d. Select **Run script**.
- e. Select the back icon (<) to return to the guided setup page tab.
- f. From the application picker, select the **Service Graph Connector for AWS** application scope.

- g. Set the Clear Cache for Data Source and Scheduled Data Imports tables task to complete by selecting **Mark as Complete**.
- h. Create new data sources and scheduled imports.
 - a. For the Create New Data Sources and Scheduled Imports task, select **Configure**.
 - b. On the form that opens in a new tab, fill in the fields.

SG-AWS Create Data Source and Scheduled Import form

Field	Description
Data source and Scheduled Import name prefix	Identifier that is used in all the data sources and scheduled import names for this AWS connection. In a multi-instance deployment, the prefix should be a short, meaningful identifier that enables you to identify a set of related data sources.
Connection and Credentials Alias	Connection alias that was created in step 11.b.
Run Scheduled Imports as User	User who will run the scheduled data import.

- c. Select **Submit**.
- d. Close the tab for the SG-AWS Create Data Source and Scheduled Import form and return to the guided setup page tab.
- e. Set the Create New Data Sources and Scheduled Imports task to complete by selecting **Mark as Complete**.
- i. Configure the scheduled imports for the new AWS instance.
 - a. For the Configure the Scheduled Imports task, select **Configure**.

- b. In the Scheduled Data Imports list that opens in a new tab, select the organization of the AWS instance that you want to configure.
- c. Select the scheduled job that you want to activate.
- d. On the Scheduled Data Import form, modify the field values for the scheduled job.
- e. Select **Execute Now**.
- f. Repeat the steps 11.i.iii to 11.i.v for each scheduled job for data import.
- g. Close the tab for the Scheduled Data Imports list and return to the guided setup page tab.
- h. Set the Configure the Scheduled Imports task to complete by selecting **Mark as Complete** in the guided setup.

When you complete the guided setup, you can configure the integration to periodically pull data from AWS. The data is saved in tables that extend from the Configuration item [cmdb_ci] table.

Availability Zone [cmdb_ci_availability_zone]

The following attributes in the Availability Zone [cmdb_ci_availability_zone] table are populated by collected data:

Attribute label	Attribute name
Name	name
Object ID	object_id

Relationships created for Availability Zone

Parent class	Relationship type	Child class
Availability Zone [cmdb_ci_availability_zone]	Contains::Contained by	Cloud Subnet [cmdb_ci_cloud_subnet]

Parent class	Relationship type	Child class
Availability Zone [cmdb_ci_availability_zone]	Contains::Contained by	Cloud Load Balancer [cmdb_ci_cloud_load_balancer]

Block Endpoint [cmdb_ci_endpoint_block]

The following attributes in the Block Endpoint [cmdb_ci_endpoint_block] table are populated by collected data:

Attribute label	Attribute name
Host	host
Name	name
Object ID	object_id

Cloud DataBase [cmdb_ci_cloud_database]

The following attributes in the Cloud DataBase [cmdb_ci_cloud_database] table are populated by collected data:

Attribute label	Attribute name
Name	name
Object ID	object_id
TCP port(s)	tcp_port
Fully qualified domain name	fqdn
Install Status	install_status
Type	type
Version	version

Relationships created for Cloud DataBase

Parent class	Relationship type	Child class
Cloud DataBase [cmdb_ci_cloud_data base]	Hosted on::Hosts	AWS Datacenter [cmdb_ci_aws_datac enter]
Cloud DataBase [cmdb_ci_cloud_data base]	Reference	Key Value [cmdb_key_value]

Cloud Function [cmdb_ci_cloud_function]

The following attributes in the Cloud Function [cmdb_ci_cloud_function] table are populated by collected data:

Attribute label	Attribute name
Name	name
Object ID	object_id
Code Size	code_size
CodeSha256	codesha256
Function Last Modified	function_last_modified
Language	language
Install Status	install_status
Version	version

Relationships created for Cloud Function

Parent class	Relationship type	Child class
Cloud Function [cmdb_ci_cloud_func tion]	Hosted on::Hosts	AWS Datacenter [cmdb_ci_aws_datac enter]

Parent class	Relationship type	Child class
Cloud Function [cmdb_ci_cloud_function]	Reference	Key Value [cmdb_key_value]

Cloud Gateway [cmdb_ci_cloud_gateway]

The following attributes in the Cloud Gateway [cmdb_ci_cloud_gateway] table are populated by collected data:

Attribute label	Attribute name
Name	name
Object ID	object_id
Fully qualified domain name	fqdn
Install Status	install_status

Relationships created for Cloud Gateway

Parent class	Relationship type	Child class
Cloud Gateway [cmdb_ci_cloud_gateway]	Hosted on::Hosts	AWS Datacenter [cmdb_ci_aws_datacenter]
Cloud Gateway [cmdb_ci_cloud_gateway]	Reference	Key Value [cmdb_key_value]

Cloud Load Balancer [cmdb_ci_cloud_load_balancer]

The following attributes in the Cloud Load Balancer [cmdb_ci_cloud_load_balancer] table are populated by collected data:

Attribute label	Attribute name
Name	name

Attribute label	Attribute name
Object ID	object_id
Install Status	install_status

Relationships created for Cloud Load Balancer

Parent class	Relationship type	Child class
Cloud Load Balancer [cmdb_ci_cloud_load_balancer]	Contains::Contained by	Compute Security Group [cmdb_ci_compute_security_group]
Cloud Load Balancer [cmdb_ci_cloud_load_balancer]	Hosted on::Hosts	AWS Datacenter [cmdb_ci_aws_datacenter]
Cloud Load Balancer [cmdb_ci_cloud_load_balancer]	Reference	Key Value [cmdb_key_value]

Cloud Mgmt Network Interface [cmdb_ci_nic]

The following attributes in the Cloud Mgmt Network Interface [cmdb_ci_nic] table are populated by collected data:

Attribute label	Attribute name
Name	name
Object ID	object_id
Public IP	public_ip
Private DNS	private_dns
Private IP	private_ip
Public DNS	public_dns

Attribute label	Attribute name
State	state
Install Status	install_status

Relationships created for Cloud Mgmt Network Interface

Parent class	Relationship type	Child class
Cloud Mgmt Network Interface [cmdb_ci_nic]	Hosted on::Hosts	AWS Datacenter [cmdb_ci_aws_datacenter]
Cloud Mgmt Network Interface [cmdb_ci_nic]	Use End Point To::Use End Point From	VNIC Endpoint [cmdb_ci_endpoint_vnic]
Cloud Mgmt Network Interface [cmdb_ci_nic]	Reference	Key Value [cmdb_key_value]

Cloud Network [cmdb_ci_network]

The following attributes in the Cloud Network [cmdb_ci_network] table are populated by collected data:

Attribute label	Attribute name
Name	name
Object ID	object_id
Install Status	install_status

Relationships created for Cloud Network

Parent class	Relationship type	Child class
Cloud Network [cmdb_ci_network]	Hosted on::Hosts	AWS Datacenter [cmdb_ci_aws_datacenter]

Parent class	Relationship type	Child class
Cloud Network [cmdb_ci_network]	Contains::Contained by	Cloud Subnet [cmdb_ci_cloud_subnet]
Cloud Network [cmdb_ci_network]	Contains::Contained by	Compute Security Group [cmdb_ci_compute_security_group]
Cloud Network [cmdb_ci_network]	Reference	Key Value [cmdb_key_value]

Cloud Object Storage [cmdb_ci_cloud_object_storage]

The following attributes in the Cloud Object Storage [cmdb_ci_cloud_object_storage] table are populated by collected data:

Attribute label	Attribute name
Name	name
Object ID	object_id
Cloud Provider	cloud_provider
Service Name	service_name
Install Status	install_status

Relationships created for Cloud Object Storage

Parent class	Relationship type	Child class
Cloud Object Storage [cmdb_ci_cloud_object_storage]	Hosted on::Hosts	AWS Datacenter [cmdb_ci_aws_datacenter]

Parent class	Relationship type	Child class
Cloud Object Storage [cmdb_ci_cloud_object_storage]	Reference	Key Value [cmdb_key_value]

Cloud Organizations [cmdb_ci_cloud_org]

The following attributes in the Cloud Organizations [cmdb_ci_cloud_org] table are populated by collected data:

Attribute label	Attribute name
Object ID	object_id
Name	name

Relationship created for Cloud Organizations

Parent class	Relationship type	Child class
Cloud Organizations [cmdb_ci_cloud_org]	Contains::Contained by	Cloud Service Account [cmdb_ci_cloud_service_account]

Cloud Service Account [cmdb_ci_cloud_service_account]

The following attributes in the Cloud Service Account [cmdb_ci_cloud_service_account] table are populated by collected data:

Attribute label	Attribute name
Is master account	is_master_account
Parent account	parent_account
Account Id	account_id
Name	name

Attribute label	Attribute name
Object ID	object_id
Datacenter Type	datacenter_type

Relationships created for Cloud Service Account

Parent class	Relationship type	Child class
Cloud Service Account [cmdb_ci_cloud_service_account]	Reference	Cloud Service Account [cmdb_ci_cloud_service_account]
Cloud Service Account [cmdb_ci_cloud_service_account]	Reference	Key Value [cmdb_key_value]

Cloud Subnet [cmdb_ci_cloud_subnet]

The following attributes in the Cloud Subnet [cmdb_ci_cloud_subnet] table are populated by collected data:

Attribute label	Attribute name
Name	name
Object ID	object_id
Available IP Count	available_ip_count
CIDR	cidr
Install Status	install_status

Relationships created for Cloud Subnet

Parent class	Relationship type	Child class
Cloud Subnet [cmdb_ci_cloud_subnet]	Contains::Contained by	Cloud Load Balancer [cmdb_ci_cloud_load_balancer]
Cloud Subnet [cmdb_ci_cloud_subnet]	Contains::Contained by	Cloud Mgmt Network Interface [cmdb_ci_nic]
Cloud Subnet [cmdb_ci_cloud_subnet]	Reference	Key Value [cmdb_key_value]

Compute Security Group [cmdb_ci_compute_security_group]

The following attributes in the Compute Security Group [cmdb_ci_compute_security_group] table are populated by collected data:

Attribute label	Attribute name
Name	name
Object ID	object_id
Install Status	install_status

Relationships created for Compute Security Group Table

Parent class	Relationship type	Child class
Compute Security Group [cmdb_ci_compute_security_group]	Hosted on::Hosts	AWS Datacenter [cmdb_ci_aws_datacenter]
Compute Security Group	Reference	Key Value [cmdb_key_value]

Parent class	Relationship type	Child class
[cmdb_ci_compute_security_group]		

Docker Container [cmdb_ci_docker_container]

The following attributes in the Docker Container [cmdb_ci_docker_container] table are populated by collected data:

Attribute label	Attribute name
Container id	container_id
Name	name
Status	status

Docker Image [cmdb_ci_docker_image]

The following attributes in the Docker Image [cmdb_ci_docker_image] table are populated by collected data:

Attribute label	Attribute name
Image id	image_id
Name	name

DynamoDB [cmdb_ci_dynamodb_table]

The following attributes in the DynamoDB [cmdb_ci_dynamodb_table] table are populated by collected data:

Attribute label	Attribute name
Name	name
Object ID	object_id
Install Status	install_status

Attribute label	Attribute name
Read Units	read_units
Write Units	write_units

Relationships created for DynamoDB Table

Parent class	Relationship type	Child class
DynamoDB [cmdb_ci_dynamodb_table]	Hosted on::Hosts	AWS Datacenter [cmdb_ci_aws_datacenter]
DynamoDB [cmdb_ci_dynamodb_table]	Reference	Key Value [cmdb_key_value]

Hardware Type [cmdb_ci_compute_template]

The following attributes in the Hardware Type [cmdb_ci_compute_template] table are populated by collected data:

Attribute label	Attribute name
Name	name
Object ID	object_id
Cores	cores
Local Storage GB	local_storage_gb
Memory MB	memory_mb
vCPUs	vcpus

Relationship created for Hardware Type

Parent class	Relationship type	Child class
Hardware Type [cmdb_ci_compute_template]	Hosted on::Hosts	AWS Datacenter [cmdb_ci_aws_datacenter]

Image [cmdb_ci_os_template]

The following attributes in the Image [cmdb_ci_os_template] table are populated by collected data:

Attribute label	Attribute name
Name	name
Object ID	object_id
Description	short_description
Environment	environment
Guest OS	guest_os
Image Source	image_source
Image Type	image_type
Root Device Type	root_device_type

Relationships created for Image

Parent class	Relationship type	Child class
Image [cmdb_ci_os_template]	Hosted on::Hosts	AWS Datacenter [cmdb_ci_aws_datacenter]
Image [cmdb_ci_os_template]	Reference	Key Value

IP Address [cmdb_ci_ip_address]

The following attributes in the IP Address [cmdb_ci_ip_address] table are populated by collected data:

Attribute label	Attribute name
Nic	nic
IP Address	ip_address

Relationship created for IP Address

Parent class	Relationship type	Child class
IP Address [cmdb_ci_ip_address]	Reference	Network Adapter [cmdb_ci_network_adapter]

Key Value [cmdb_key_value]

The following attributes in the Key Value [cmdb_key_value] table are populated by collected data:

Attribute label	Attribute name
Key	key
Value	Value

Kubernetes Cluster [cmdb_ci_kubernetes_cluster]

The following attributes in the Kubernetes Cluster [cmdb_ci_kubernetes_cluster] table are populated by collected data:

Attribute label	Attribute name
Name	name
Port	port
Install status	install_status

Relationships created for Kubernetes Cluster

Parent class	Relationship type	Child class
Kubernetes Cluster [cmdb_ci_kubernetes_cluster]	Contains::Contained by	Kubernetes Pod [cmdb_ci_kubernetes_pod]
Kubernetes Cluster [cmdb_ci_kubernetes_cluster]	Cluster of::Cluster	Kubernetes Node [cmdb_ci_kubernetes_node]
Kubernetes Cluster [cmdb_ci_kubernetes_cluster]	Managed by::Manages	Server [cmdb_ci_server]
Kubernetes Cluster [cmdb_ci_kubernetes_cluster]	Hosted on::Hosts	AWS Datacenter [cmdb_ci_aws_datacenter]
Kubernetes Cluster [cmdb_ci_kubernetes_cluster]	Contains::Contained by	Kubernetes Namespace [cmdb_ci_kubernetes_namespace]
Kubernetes Cluster [cmdb_ci_kubernetes_cluster]	Contains::Contained by	Kubernetes Service [cmdb_ci_kubernetes_service]
Kubernetes Cluster [cmdb_ci_kubernetes_cluster]	Reference	Key Value [cmdb_key_value]

Kubernetes DaemonSet [cmdb_ci_kubernetes_daemonset]

The following attributes in the Kubernetes DaemonSet [cmdb_ci_kubernetes_daemonset] table are populated by collected data:

Attribute label	Attribute name
Kubernetes UID	k8s_uid

Attribute label	Attribute name
Name	name
Namespace	namespace
SelfLink	self_link

Relationship created for Kubernetes DaemonSet

Parent class	Relationship type	Child class
Kubernetes DaemonSet [cmdb_ci_kubernetes_daemonset]	Hosted on::Hosts	Kubernetes Cluster [cmdb_ci_kubernetes_cluster]

Kubernetes Deployment [cmdb_ci_kubernetes_deployment]

The following attributes in the Kubernetes Deployment [cmdb_ci_kubernetes_deployment] table are populated by collected data:

Attribute label	Attribute name
Kubernetes UID	k8s_uid
Name	name
Namespace	namespace
Available Replicas	available_replicas
SelfLink	self_link

Relationship created for Kubernetes Deployment

Parent class	Relationship type	Child class
Kubernetes Deployment	Hosted on::Hosts	Kubernetes Cluster [cmdb_ci_kubernetes_cluster]

Parent class	Relationship type	Child class
[cmdb_ci_kubernetes_deployment]		

Kubernetes Namespace [cmdb_ci_kubernetes_namespace]

The following attributes in the Kubernetes Namespace [cmdb_ci_kubernetes_namespace] table are populated by collected data:

Attribute label	Attribute name
Kubernetes UID	k8s_uid
Namespace	namespace

Kubernetes Node [cmdb_ci_kubernetes_node]

The following attributes in the Kubernetes Node [cmdb_ci_kubernetes_node] table are populated by collected data:

Attribute label	Attribute name
Kubernetes UID	k8s_uid
Name	name

Kubernetes Pod [cmdb_ci_kubernetes_pod]

The following attributes in the Kubernetes Pod [cmdb_ci_kubernetes_pod] table are populated by collected data:

Attribute label	Attribute name
Kubernetes UID	k8s_uid
Name	name
Namespace	namespace
Resource version	resource_version

Relationship created for Kubernetes Pod

Parent class	Relationship type	Child class
Kubernetes Pod [cmdb_ci_kubernetes_pod]	Contains::Contained by	Docker Container [cmdb_ci_docker_container]
Kubernetes Pod [cmdb_ci_kubernetes_pod]	Contains::Contained by	Kubernetes Volume [cmdb_ci_kubernetes_volume]
Kubernetes Pod [cmdb_ci_kubernetes_pod]	Contains::Contained by	Docker Image [cmdb_ci_docker_image]

Kubernetes Service [cmdb_ci_kubernetes_service]

The following attributes in the Kubernetes Service [cmdb_ci_kubernetes_service] table are populated by collected data:

Attribute label	Attribute name
Kubernetes UID	k8s_uid
Name	name
Namespace	namespace
IP Address	ip_address

Kubernetes Volume [cmdb_ci_kubernetes_volume]

The following attributes in the Kubernetes Volume [cmdb_ci_kubernetes_volume] table are populated by collected data:

Attribute label	Attribute name
Kubernetes UID	k8s_uid
Mount Path	mount_path

Attribute label	Attribute name
Name	name
Namespace	namespace
Volume ID	volume_id

AWS Datacenter [cmdb_ci_aws_datacenter]

The following attributes in the AWS Datacenter [cmdb_ci_aws_datacenter] table are populated by collected data:

Attribute label	Attribute name
Name	name
Object ID	object_id
Region	region

Relationships created for AWS Datacenter

Parent class	Relationship type	Child class
AWS Datacenter [cmdb_ci_aws_datacenter]	Contains::Contained by	Availability Zone [cmdb_ci_availability_zone]
AWS Datacenter [cmdb_ci_aws_datacenter]	Hosted on::Hosts	Cloud Service Account [cmdb_ci_cloud_service_account]

Network Adapter [cmdb_ci_network_adapter]

The following attributes in the Network Adapter [cmdb_ci_network_adapter] table are populated by collected data:

Attribute label	Attribute name
IP Address	ip_address
MAC Address	mac_address
Configuration Item	cmdb_ci
Name	name

Relationship created for Network Adapter

Parent class	Relationship type	Child class
Network Adapter [cmdb_ci_network_adapter]	Reference	Server [cmdb_ci_server], Linux Server [cmdb_ci_linux_server] , or Windows Server [cmdb_ci_win_server]

Server [cmdb_ci_server]

The following attributes in the Server [cmdb_ci_server] table are populated by collected data:

Attribute label	Attribute name
Name	name
Class	sys_class_name
CPU core count	cpu_core_count
CPU core thread	cpu_core_thread
CPU count	cpu_count
CPU name	cpu_name
CPU speed (MHz)	cpu_speed

Attribute label	Attribute name
Disk Space (GB)	disk_space
DNS Domain	dns_domain
Install Status	install_status
Is Virtual	virtual
Operating System	os
Operational status	operational_status
OS Version	os_version
RAM (MB)	ram

Note: If the AWS Systems Manager (SSM) service isn't enabled, the connector populates the server records in the Server [cmdb_ci_server] class. If the AWS SSM service is enabled, then based on the platform type obtained through the SSM service, the server records are populated in either the Linux Server [cmdb_ci_linux_server] class or the Windows Server [cmdb_ci_win_server] class. The Server [cmdb_ci_server] class is the parent class of the Linux Server [cmdb_ci_linux_server] and the Windows Server [cmdb_ci_win_server] classes.

Relationships created for Server

Parent class	Relationship type	Child class
Server [cmdb_ci_server]	Virtualized by::Virtualizes	Virtual Machine Instance [cmdb_ci_vm_instance]
Server [cmdb_ci_server]	Owns::Owned by	IP Address [cmdb_ci_ip_address]

Parent class	Relationship type	Child class
Server [cmdb_ci_server]	Owns::Owned by	Network Adapter [cmdb_ci_network_adapter]
Server [cmdb_ci_server]	Reference	Software Installation [cmdb_sam_sw_install]
Server [cmdb_ci_server]	Reference	Key Value [cmdb_key_value]

Software [cmdb_ci_spkg]

The following attributes in the Software [cmdb_ci_spkg] table are populated by collected data when the Software Asset Management (SAM) application isn't installed:

Attribute label	Attribute name
Key	key
Discovery source	discovery_source
Name	name
Version	version
Manufacturer	manufacturer

Relationship created for Software

Parent class	Relationship type	Child class
Software [cmdb_ci_spkg]	Reference	Software Instance [cmdb_software_instance]

Software Installation [cmdb_sam_sw_install]

The following attributes in the Software Installation [cmdb_sam_sw_install] table are populated by collected data when the SAM application is installed:

Attribute label	Attribute name
Display name	display_name
Publisher	publisher
Version	version
Discovery source	discovery_source

Software Instance [cmdb_software_instance]

The following attributes in the Software Instance [cmdb_software_instance] table are populated by collected data when the SAM application isn't installed:

Attribute label	Attribute name
Name	name
Installed on	installed_on
Install date	install_date

Relationship created for Software Instance

Parent class	Relationship type	Child class
Software Instance [cmdb_software_instance]	Reference	Server [cmdb_ci_server]

Storage Mapping [cmdb_ci_storage_mapping]

The following attributes in the Storage Mapping [cmdb_ci_storage_mapping] table are populated by collected data:

Attribute label	Attribute name
Name	name
Object ID	object_id
Mount Point	mount_point

Relationship created for Storage Mapping

Parent class	Relationship type	Child class
Storage Mapping [cmdb_ci_storage_mapping]	Use End Point To::Use End Point From	Block Endpoint [cmdb_ci_endpoint_block]

Storage Volume [cmdb_ci_storage_volume]

The following attributes in the Storage Volume [cmdb_ci_storage_volume] table are populated by collected data:

Attribute label	Attribute name
Object ID	object_id
Volume ID	volume_id
Name	name
Size	size
Size bytes	size_bytes
State	state
Install Status	install_status
Storage Type	storage_type

Relationships created for Storage Volume

Parent class	Relationship type	Child class
Storage Volume [cmdb_ci_storage_volume]	Hosted on::Hosts	AWS Datacenter [cmdb_ci_aws_datacenter]
Storage Volume [cmdb_ci_storage_volume]	Reference	Key Value [cmdb_key_value]

Storage Volume Snapshot [cmdb_ci_storage_vol_snapshot]

The following attributes in the Storage Volume Snapshot [cmdb_ci_storage_vol_snapshot] table are populated by collected data:

Attribute label	Attribute name
Name	name
Object ID	object_id

Relationship created for Storage Volume Snapshot

Parent class	Relationship type	Child class
Storage Volume Snapshot [cmdb_ci_storage_vol_snapshot]	Hosted on::Hosts	AWS Datacenter [cmdb_ci_aws_datacenter]

Virtual Machine Instance [cmdb_ci_vm_instance]

The following attributes in the Virtual Machine Instance [cmdb_ci_vm_instance] table are populated by collected data:

Attribute label	Attribute name
Name	name

Attribute label	Attribute name
Object ID	object_id
CPUs	cpus
Disks	disks
Disks size (GB)	disks_size
IP Address	ip_address
Memory (MB)	memory
Monitor	monitor
Network adapters	nics
Placement Group ID	placement_group_id
State	state
Install Status	install_status
VM Instance ID	vm_inst_id
Operational status	operational_status

Relationships created for Virtual Machine Instance

Parent class	Relationship type	Child class
Virtual Machine Instance [cmdb_ci_vm_instance]	Hosted on::Hosts	AWS Datacenter [cmdb_ci_aws_datacenter]
Virtual Machine Instance [cmdb_ci_vm_instance]	Contains::Contained by	Storage Mapping [cmdb_ci_storage_mapping]

Parent class	Relationship type	Child class
Virtual Machine Instance [cmdb_ci_vm_instance]	Provisioned From::Provisioned	Image [cmdb_ci_os_template]
Virtual Machine Instance [cmdb_ci_vm_instance]	Provisioned From::Provisioned	Hardware Type [cmdb_ci_compute_template]
Virtual Machine Instance [cmdb_ci_vm_instance]	Reference	Key Value [cmdb_key_value]
Virtual Machine Instance [cmdb_ci_vm_instance]	Use End Point To::Use End Point From	Storage Volume [cmdb_ci_storage_volume]
Virtual Machine Instance [cmdb_ci_vm_instance]	Use End Point To::Use End Point From	Cloud Mgmt Network Interface [cmdb_ci_nic]

VNIC Endpoint [cmdb_ci_endpoint_vnic]

The following attributes in the VNIC Endpoint [cmdb_ci_endpoint_vnic] table are populated by collected data:

Attribute label	Attribute name
Host	host
Name	name
Object ID	object_id

Service Graph Connector for Microsoft Azure (1.4.0)

Use the Service Graph Connector for Microsoft Azure to pull data from Microsoft Azure into your CMDB.

Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

Supported versions

Supported ServiceNow versions:

- San Diego
- Tokyo
- Utah

Use cases

The following are examples on how you can use the Service Graph Connector:

- Visibility into cloud resources, relationships, and state in near real-time.
- Service ITAM/SAM outcomes through deep discovery of applications.

Guided setup

The guided setup for the Service Graph Connector for Microsoft Azure provides an organized sequence of tasks to configure the integration on your instance. To access the guided setup, see [Configure guided setup](#).

CMDB integrations dashboard

The Integration Commons for CMDB store app provides a dashboard with a central view of the status, processing results, and processing errors of all installed integrations. You can see metrics for all integration runs. You can filter the view to a specific CMDB integration, a specific time duration, or a specific integration run. For more details about monitoring Microsoft

Azure integrations in the CMDB Integrations Dashboard, see [Using the CMDB Integrations Dashboard](#).

Data mapping

Data from the Azure data sources is mapped and transformed into the ServiceNow CMDB Configuration Item (CI) class definitions using the Robust Transform Engine (RTE). Data is inserted into the ServiceNow® CMDB using the Identification and Reconciliation Engine (IRE).

You can use the IntegrationHub ETL app to view the data maps. See [IntegrationHub ETL \(3.2\)](#) for more information.

The Azure data sources include the following:

- SG-Azure Availability Zone
- SG-Azure Datacenters
- SG-Azure Hardware Template
- SG-Azure Load Balancers
- SG-Azure Network
- SG-Azure Network Interface
- SG-Azure Public IP Address
- SG-Azure Resource Group
- SG-Azure Security Group
- SG-Azure Server Config Data
- SG-Azure Software
- SG-Azure Storage Accounts
- SG-Azure Storage Volume
- SG-Azure Subscriptions
- SG-Azure Virtual Machines
- SG-Azure SQL

When you complete the guided setup, you can configure the integration to periodically pull data from Azure. The data is loaded into staging tables and then inserted into the following target tables:

- Availability Zone [cmdb_ci_availability_zone]
- Azure Datacenter [cmdb_ci_azure_datacenter]
- Cloud LB Public IP Address [cmdb_ci_cloud_lb_ipaddress]
- Cloud Load Balancer [cmdb_ci_cloud_load_balancer]
- Cloud Mgmt Network Interfaces [cmdb_ci_nic]
- Cloud Network [cmdb_ci_network]
- Cloud Public IP Address [cmdb_ci_cloud_public_ipaddress]
- Cloud Service Account [cmdb_ci_cloud_service_account]
- Cloud Storage Account [cmdb_ci_cloud_storage_account]
- Cloud Subnet [cmdb_ci_cloud_subnet]
- Compute Security Groups [cmdb_ci_compute_security_group]
- Hardware Type [cmdb_ci_compute_template]
- Image [cmdb_ci_os_template]
- Key Value [cmdb_key_value]
- Linux Server [cmdb_ci_linux_server]
- Resource Group [cmdb_ci_resource_group]
- Servers [cmdb_ci_server]
- Software [cmdb_ci_spkg]
- Software Installation [cmdb_sam_sw_install]
- Software Instance [cmdb_software_instance]
- Storage Volume [cmdb_ci_storage_volume]
- Virtual Server [cmdb_ci_vm_instance]

- Windows Server [cmdb_ci_win_server]

For more information on where data is saved when pulling data from Azure, see [CMDB classes targeted](#).

Set up data sources and scheduled import jobs to pull in data from Azure into your CMDB.

Before you begin

Dependencies and requirements:

- The [Integration Commons for CMDB](#) store app, which is automatically installed.
- The [CMDB CI Class Models](#) store app store app, which is automatically installed.
- Discovery Core plugin (com.snc.discovery.core), which is automatically installed by Discovery.
- The Datastream Action plugin (com.glide.hub.action_type.datastream), which is automatically installed.
- The ITOM Discovery License plugin (com.snc.item.discovery.license). You must activate this plugin.
- ITOM Licensing plugin (com.snc.item.license). For more information, see [Request Discovery](#).

Starting with the San Diego release, embedded help content will not be visible on the default Polaris theme in the Next Experience. You must activate the embedded help content by completing the following steps:

1. Select a task section in the guided setup, and then click **Configure**.
2. In the menu bar, click the help icon (?

Role required: admin

About this task

The connector uses the Azure Management APIs for the complete pull of data from Azure. However, to pull delta changes from Azure, the Azure Resource Graph APIs are used. The domain name system (DNS) is Microsoft Azure Management, but the path is a resource graph.

For more information on the Azure setup instructions, see the [Service Graph Connector for Azure - Overview](#) article on the ServiceNow Community site.

Procedure

1. Navigate to **Service Graph Connectors > Azure > Setup**.
2. On the Getting started page, select **Get Started**.
3. Create data sources and scheduled imports for the new connection.
 - a. On the Service Graph Connector for Microsoft Azure page, in the Update Data Sources and Scheduled Imports Access, select the task **Update Scheduled Data Import Access**.
 - b. On the next page, in the Update Scheduled Data Import Access section, select **Configure** and do the following:
 - a. To edit the record, select **Global** from the Scope menu.
 - b. Under the **Application Access** tab, select the **Can create**, **Can update**, and **Can delete** check boxes.
 - c. Save the record.
 - d. From the Scope menu, select **Service Graph Connector for Microsoft Azure**.
 - e. In the Help task bar, click **Mark as Complete**.
 - f. Repeat these steps in the Update Data Source Access section with the Data Source table.
- c. Clear the cache for the new connection.
 - a. In the Clear Cache for Datasource and Import set section, select **Configure**.

b. Switch to **Global** in the Scope menu.

c. Enter the following script.

```
GlideTableManager.invalidateTable("sys_data_sou  
rce");  
    GlideCacheManager.flushTable("sys_data_sou  
rce");  
  
    GlideTableManager.invalidateTable("schedu  
led_import_set");  
    GlideCacheManager.flushTable("scheduled_im  
port_set");  
  
    GlideTableManager.invalidateTable("sys_db  
_object");  
    GlideCacheManager.flushTable("sys_db_object  
");
```

d. Select **Run script**.

e. From the Scope menu, select **Service Graph Connector for Microsoft Azure**.

f. Click **Mark as Complete**.

4. Create a connection to import hardware configuration items (CIs) from the Azure client application.

Note: Ensure that you have the **User.Read** permission on the Microsoft Graph API for the hardware import.

a. Obtain the OAuth credentials from your Azure administrator. Make a note of the following details:

- Application (client) ID
- Client Secret
- Directory (tenant) ID
- Connection URL

Note: After getting the OAuth credentials, in the guided setup for Service Graph Connector for Microsoft Azure, go to the Create connection for the hardware import section of the Service Graph Connector for Microsoft Azure page and set the Get the OAuth credentials task to complete by clicking **Mark as Complete**.

- b. Configure your Azure hardware connection and credentials.
 - a. In the Create connection for the hardware import section of the Service Graph Connector for Microsoft Azure page, select **Continue**.
 - b. For the Create or Edit connection task, select **Configure**.
 - c. On the Connections page of the Flow Designer, select **Configure** for the **SG-Azure Hardware Connection** connection that is available by default for the hardware import.
- You can create multiple connections by clicking **Add Connection**.
- d. On the form, review and modify the fields.

Configure Connection form

Field	Description
Connection Information	
Connection Name	Name to uniquely identify the hardware connection record. For example, SG-Azure Hardware Connection .
Connection URL	Base URL to connect to your Azure client application.

Field	Description
	<p>Note: This field is automatically set to the URL to connect to the application. Leave the field value as is.</p>
Credential Information	
OAuth Client ID	Application (client) ID of your Azure client application as described in step 4.a .
OAuth Client Secret	Client Secret of your Azure client application as described in step 4.a .
OAuth Token URL	<p>Token URL of your Azure client application. Based on the region of your Azure client application, enter the token URL in one of the following formats:</p> <ul style="list-style-type: none">• Global <code>https://login.microsoftonline.com/<tenantid>/oauth2/v2.0/token</code>• US Government <code>https://login.microsoftonline.us/<tenantid>/oauth2/v2.0/token</code>• China <code>https://login.partner.microsoftonline.</code>

Field	Description
	<p>cn/<tenantid>/oauth2/v2.0/token</p> <ul style="list-style-type: none">Germany <p><a href="https://login.microsoftonline.de/<tenantid>/oauth2/v2.0/token">https://login.microsoftonline.de/<tenantid>/oauth2/v2.0/token</p> <p>Where <tenantid> is the tenant ID of your Azure client application as described in step 4.a.</p>

- e. Select **Configure and Get OAuth Token**.
- f. When the OAuth token flow is successfully completed, return to the Create connection for the hardware import task page using the back button for your browser.
- g. Set the Create or Edit connection task to complete by clicking **Mark as Complete**.
- c. Test the Microsoft Graph API connection to import hardware data from the Azure client application.
 - a. For the Test Connection task, select **Configure**.
 - b. Select the data source associated with the newly created connection in the **Name** column of the Data Sources list.
 - c. Click the **Test Load 20 Records** related link.
 - d. When the state changes to **Complete**, return to the setup by clicking **Back to Guided Setup** in the Help panel.
 - e. Set the Test Connection task to complete by clicking **Mark as Complete**.
 - d. Review the scheduled data imports configuration.
 - a. For the Set up scheduled import jobs task, select **Configure**.

- b. Select the SG-Azure Subscriptions scheduled job.
- c. On the Scheduled Data Import form, verify the field values for the scheduled job.

For more information, see [Schedule a data import](#).

- d. Click **Execute Now**.
- e. Set the Set up scheduled import jobs task to complete by clicking **Mark as Complete** in the Help panel.

5. Create a connection to import software information from the Azure client application.

Note: Ensure that you have the **Data.Read** permission on the Log Analytics API for the software import.

- a. Obtain the OAuth credentials and set up the Log Analytics workspace in the Azure.
 - a. Ensure that you have the OAuth credentials from step [4.a](#).
 - b. Configure the Log Analytics workspace in the Azure client application.
 - Use an existing workspace, if available.
 - Create another workspace.

For more information, see [Create Log Analytics workspace](#) in the Azure documentation.

Note: Make a note of the Workspace ID.

- c. Create an automation account in the Azure client application.

For more information, see [Quickstart: Create an Automation account using the Azure portal](#) in the Azure documentation.

- d. Enable change tracking and inventory from the automation account.

For more information, see [Enable Change Tracking and Inventory from an Automation account](#) in the Azure documentation.

Note: After obtaining the OAuth credentials, in the guided setup for Service Graph Connector for Microsoft Azure, go to the Create connection for the software import section of the Service Graph Connector for Microsoft Azure page and set the Get the OAuth credentials task to complete by clicking **Mark as Complete**.

- b. Configure your Azure software connection and credentials.
 - a. In the Create connection for the software import section of the Service Graph Connector for Microsoft Azure page, select **Continue**.
 - b. For the Create or Edit connection task, select **Configure**.
 - c. On the Connections page of the Flow Designer, select **Configure** for the **SG-Azure log analytics connection** connection that is available by default for the software import.

You can create multiple connections by clicking **Add Connection**.

- d. On the form, review and modify the fields.

Configure Connection form

Field	Description
Connection Information	
Software Connection Name	Name to uniquely identify the software connection record. For example, SG-Azure log analytics connection.
Hardware Connection Name	Name of the hardware connection associated with

Field	Description
	<p>the software as described in step 4.b.iv.</p> <p>You add a hardware connection name to associate the software connection with the respective hardware.</p>
Connection URL	<p>Base URL to connect to the Log Analytics workspace in the following format:</p> <pre data-bbox="894 931 1269 1030"><code>https://api.loganalyticss.io/v1/workspaces/<workspace_id></code></pre> <p>Where <workspace_id> is the ID of the Log Analytics workspace as described in step 5.a.ii.</p> <p>Note: This field is automatically set to the URL to connect to the Log Analytics workspace. Replace the <workspace_id> variable in the auto-generated URL with the workspace ID of your Log Analytics workspace.</p>
Credential Information	
OAuth Client ID	Application (client) ID of your Azure client application as described in step 4.a .

Field	Description
OAuth Client Secret	Client Secret of your Azure client application as described in step 4.a.
OAuth Token URL	<p>Token URL of your Azure client application. Based on the region of your Azure client application, enter the token URL in one of the following formats:</p> <ul style="list-style-type: none"> • Global <code>https://login.microsoftonline.com/<tenantid>/oauth2/v2.0/token</code> • US Government <code>https://login.microsoftonline.us/<tenantid>/oauth2/v2.0/token</code> • China <code>https://login.partner.microsoftonline.cn/<tenantid>/oauth2/v2.0/token</code> • Germany <code>https://login.microsoftonline.de/<tenantid>/oauth2/v2.0/token</code> <p>Where <tenantid> is the tenant ID of your Azure client</p>

Field	Description
	application as described in step 4.a.

- e. Select **Configure and Get OAuth Token**.
 - f. When the OAuth token flow is successfully completed, return to the Create connection for the software import task page using the back button for your browser.
 - g. Set the Create or Edit connection task to complete by clicking **Mark as Complete**.
 - c. Test the Log Analytics API connection to import software data from the Azure client application.
 - a. For the Test Connection task, select **Configure**.
 - b. Select the data source associated with the newly created connection in the **Name** column of the Data Sources list.
 - c. Click the **Test Load 20 Records** related link.
 - d. When the state changes to **Complete**, return to the setup by clicking **Back to Guided Setup** in the Help panel.
 - e. Set the Test Connection task to complete by clicking **Mark as Complete**.
 - d. Review the scheduled data imports configuration.
 - a. For the Set up scheduled import jobs task, select **Configure**.
 - b. Select the SG-Azure Subscriptions scheduled job.
 - c. On the Scheduled Data Import form, verify the field values for the scheduled job.
- For more information, see [Schedule a data import](#).
- d. Click **Execute Now**.
 - e. Set the Set up scheduled import jobs task to complete by clicking **Mark as Complete** in the Help panel.

Result

The data from Azure is pulled into your CMDB.

Note: You can use Azure policies to add multiple virtual machines (VMs) of different subscriptions to a single Workspace. For more information, see [Configure a policy](#).

When you complete the guided setup, you can configure the integration to periodically pull data from Microsoft Azure. The data is saved in tables that extend from the Configuration item [cmdb_ci] table.

Availability Zone [cmdb_ci_availability_zone]

The following attributes in the Availability Zone [cmdb_ci_availability_zone] table are populated by collected data.

Attribute label	Attribute name
Name	name
Object ID	object_id
State	state
Status	install_status

Relationship created for Availability Zone

Parent class	Relationship type	Child class
Availability Zone [cmdb_ci_availability_zone]	Reference	Key Value [cmdb_key_value]

Cloud DataBase [cmdb_ci_cloud_database]

The following attributes in the Cloud DataBase [cmdb_ci_cloud_database] table are populated by collected data.

Attribute label	Attribute name
Name	name
Object ID	object_id
Fully qualified domain name	fqdn
State	state
Status	install_status
Type	type
Version	version

Relationships created for Cloud DataBase

Parent class	Relationship type	Child class
Cloud DataBase [cmdb_ci_cloud_data base]	Hosted on::Hosts	Azure Datacenter [cmdb_ci_azure_data center]
Cloud DataBase [cmdb_ci_cloud_data base]	Reference	Key Value [cmdb_key_value]

Cloud LB IPAddress [cmdb_ci_cloud_lb_ipaddress]

The following attributes in the Cloud LB IPAddress [cmdb_ci_cloud_lb_ipaddress] table are populated by collected data.

Attribute label	Attribute name
Name	name
Object ID	object_id
IPAddress Type	ipaddress_type

Attribute label	Attribute name
Status	install_status

Cloud Load Balancer [cmdb_ci_cloud_load_balancer]

The following attributes in the Cloud Load Balancer [cmdb_ci_cloud_load_balancer] table are populated by collected data.

Attribute label	Attribute name
Name	name
Object ID	object_id
Canonical Hosted Zone Name	canonical_hosted_zone_name
DNS Name	dns_name
Fully qualified domain name	fqdn
State	state

Relationships created for Cloud Load Balancer

Parent class	Relationship type	Child class
Cloud Load Balancer [cmdb_ci_cloud_load_balancer]	Hosted on::Hosts	Azure Datacenter [cmdb_ci_azure_data_center]
Cloud Load Balancer [cmdb_ci_cloud_load_balancer]	Owns::Owned by	Cloud LB IPAddress [cmdb_ci_cloud_lb_ip_address]
Cloud Load Balancer [cmdb_ci_cloud_load_balancer]	Reference	Key Value [cmdb_key_value]

Key Value [cmdb_key_value]

The following attributes in the Key Value [cmdb_key_value] table are populated by collected data.

Attribute label	Attribute name
Key	key
Value	value

Cloud Mgmt Network Interface [cmdb_ci_nic]

The following attributes in the Cloud Mgmt Network Interface [cmdb_ci_nic] table are populated by collected data.

Attribute label	Attribute name
MAC Address	mac_address
Name	name
Object ID	object_id
Public IP	public_ip
Private IP	private_ip
Public DNS	public_dns
State	state
Static	is_static

Relationships created for Cloud Mgmt Network Interface

Parent class	Relationship type	Child class
Cloud Mgmt Network Interface [cmdb_ci_nic]	Hosted on::Hosts	Azure Datacenter [cmdb_ci_azure_data_center]

Parent class	Relationship type	Child class
Cloud Mgmt Network Interface [cmdb_ci_nic]	Contains::Contained by	Cloud Public IP Address [cmdb_ci_cloud_public_ipaddress]
Cloud Mgmt Network Interface [cmdb_ci_nic]	Reference	Key Value [cmdb_key_value]

Cloud Network [cmdb_ci_network]

The following attributes in the Cloud Network [cmdb_ci_network] table are populated by collected data.

Attribute label	Attribute name
Name	name
Object ID	object_id
Cidr	cidr
State	state

Relationships created for Cloud Network

Parent class	Relationship type	Child class
Cloud Network [cmdb_ci_network]	Hosted on::Hosts	Azure Datacenter [cmdb_ci_azure_data_center]
Cloud Network [cmdb_ci_network]	Contains::Contained by	Compute Security Group [cmdb_ci_compute_security_group]
Cloud Network [cmdb_ci_network]	Contains::Contained by	Cloud Subnet [cmdb_ci_cloud_subnet]

Parent class	Relationship type	Child class
Cloud Network [cmdb_ci_network]	Reference	Key Value [cmdb_key_value]

Linux Server [cmdb_ci_linux_server]

The following attributes in the Linux Server [cmdb_ci_linux_server] table are populated by collected data.

Attribute label	Attribute name
Name	name
Object ID	object_id
Serial number	serial_number
CPU core count	cpu_core_count
Disk space (GB)	disk_space
Is Virtual	virtual
Operating System	os
OS Version	os_version
RAM (MB)	ram
Status	install_status

Relationships created for Linux Server

Parent class	Relationship type	Child class
Linux Server [cmdb_ci_linux_server]	Virtualized by::Virtualizes	Virtual Machine Instance [cmdb_ci_vm_instance]

Parent class	Relationship type	Child class
Cloud Network [cmdb_ci_network]	Contains::Contained by	Compute Security Group [cmdb_ci_compute_security_group]
Cloud Network [cmdb_ci_network]	Contains::Contained by	Cloud Subnet [cmdb_ci_cloud_subnet]
Cloud Network [cmdb_ci_network]	Reference	Key Value [cmdb_key_value]

Cloud Public IP Address [cmdb_ci_cloud_public_ipaddress]

The following attributes in the Cloud Public IP Address [cmdb_ci_cloud_public_ipaddress] table are populated by collected data.

Attribute label	Attribute name
Name	name
Object ID	object_id
Public DNS	public_dns
Public IP Address	public_ip_address
Status	install_status

Relationships created for Cloud Public IP Address

Parent class	Relationship type	Child class
Cloud Public IP Address [cmdb_ci_cloud_public_ipaddress]	Hosted on::Hosts	Azure Datacenter [cmdb_ci_azure_data_center]

Parent class	Relationship type	Child class
Cloud Public IP Address [cmdb_ci_cloud_public_ipaddress]	Reference	Key Value [cmdb_key_value]

Cloud Service Account [cmdb_ci_cloud_service_account]

The following attributes in the Cloud Service Account [cmdb_ci_cloud_service_account] table are populated by collected data.

Attribute label	Attribute name
Account Id	account_id
Name	name
Object ID	object_id
Datacenter Type	datacenter_type

Relationship created for Cloud Service Account

Parent class	Relationship type	Child class
Cloud Service Account [cmdb_ci_cloud_service_account]	Reference	Key Value [cmdb_key_value]

Cloud Storage Account [cmdb_ci_cloud_storage_account]

The following attributes in the Cloud Storage Account [cmdb_ci_cloud_storage_account] table are populated by collected data.

Attribute label	Attribute name
Name	name

Attribute label	Attribute name
Object ID	object_id
Sku Name	sku_name
State	state

Relationships created for Cloud Storage Account

Parent class	Relationship type	Child class
Cloud Storage Account [cmdb_ci_cloud_storage_account]	Hosted on::Hosts	Azure Datacenter [cmdb_ci_azure_data_center]
Cloud Storage Account [cmdb_ci_cloud_storage_account]	Reference	Key Value [cmdb_key_value]

Cloud Subnet [cmdb_ci_cloud_subnet]

The following attributes in the Cloud Subnet [cmdb_ci_cloud_subnet] table are populated by collected data.

Attribute label	Attribute name
Name	name
Object ID	object_id
CIDR	cidr

Relationship created for Cloud Subnet

Parent class	Relationship type	Child class
Cloud Subnet [cmdb_ci_cloud_subnet]	Reference	Key Value [cmdb_key_value]

Compute Security Group [cmdb_ci_compute_security_group]

The following attributes in the Compute Security Group [cmdb_ci_compute_security_group] table are populated by collected data.

Attribute label	Attribute name
Name	name
Object ID	object_id
State	state

Relationships created for Compute Security Group

Parent class	Relationship type	Child class
Compute Security Group [cmdb_ci_compute_security_group]	Hosted on::Hosts	Azure Datacenter [cmdb_ci_azure_data_center]
Compute Security Group [cmdb_ci_compute_security_group]	Reference	Key Value [cmdb_key_value]

Hardware Type [cmdb_ci_compute_template]

The following attributes in the Hardware Type [cmdb_ci_compute_template] table are populated by collected data.

Attribute label	Attribute name
Name	name
Object ID	object_id
Class	sys_class_name
Cores	cores

Attribute label	Attribute name
Logical Storage GB	local_storage_gb
Memory MB	memory_mb
vCPUs	vcpus

Note: When the Cloud Hardware Type class extension is enabled, the **Class** attribute is set to **Cloud Hardware Type**. Else, the attribute is set to **Hardware Type**.

As a user with the admin role, you can enable the Cloud Hardware Type class extension by setting the value of the **use a single hardware type for cloud data centers** property (`sn_itom_pattern.use_a_single_hw_type_for_cdcs`) to true. For more information, see the [Service Graph Connector For Microsoft Azure - Migrating to a new hardware type model \[KB1288455\]](#) article in the Now Support Knowledge Base.

Relationship created for Hardware Type

Parent class	Relationship type	Child class
Hardware Type [cmdb_ci_compute_template]	Hosted on::Hosts	Azure Datacenter [cmdb_ci_azure_data_center]

Image [cmdb_ci_os_template]

The following attributes in the Image [cmdb_ci_os_template] table are populated by collected data.

Attribute label	Attribute name
Name	name
Object ID	object_id
Guest OS	guest_os

Attribute label	Attribute name
Version	version
Vendor	vendor

Relationship created for Image

Parent class	Relationship type	Child class
Image [cmdb_ci_os_template]	Hosted on::Hosts	Azure Datacenter [cmdb_ci_azure_data_center]

Software Installation [cmdb_sam_sw_install]

The following attributes in the Software Installation [cmdb_sam_sw_install] table are populated by collected data.

Attribute label	Attribute name
Display name	display_name
Publisher	publisher
Version	version
Discovery source	discovery_source

Azure Datacenter [cmdb_ci_azure_datacenter]

The following attributes in the Azure Datacenter [cmdb_ci_azure_datacenter] table are populated by collected data.

Attribute label	Attribute name
Name	name
Object ID	object_id
Region	region

Attribute label	Attribute name
Status	install_status

Relationships created for Azure Datacenter

Parent class	Relationship type	Child class
Azure Datacenter [cmdb_ci_azure_data_center]	Hosted on::Hosts	Cloud Service Account [cmdb_ci_cloud_service_account]
Azure Datacenter [cmdb_ci_azure_data_center]	Contains::Contained by	Resource Group [cmdb_ci_resource_group]
Azure Datacenter [cmdb_ci_azure_data_center]	Contains::Contained by	Availability Zone [cmdb_ci_availability_zone]

Resource Group [cmdb_ci_resource_group]

The following attributes in the Resource Group [cmdb_ci_resource_group] table are populated by collected data.

Attribute label	Attribute name
Name	name
Object ID	object_id
State	state
Status	install_status

Relationships created for Resource Group

Parent class	Relationship type	Child class
Resource Group [cmdb_ci_resource_group]	Contains::Contained by	Virtual Machine Instance [cmdb_ci_vm_instance]
Resource Group [cmdb_ci_resource_group]	Contains::Contained by	Cloud Load Balancer [cmdb_ci_cloud_load_balancer]
Resource Group [cmdb_ci_resource_group]	Contains::Contained by	Image [cmdb_ci_os_template]
Resource Group [cmdb_ci_resource_group]	Contains::Contained by	Storage Volume [cmdb_ci_storage_volume]
Resource Group [cmdb_ci_resource_group]	Reference	Key Value [cmdb_key_value]

Serial Number [cmdb_serial_number]

The following attributes in the Serial Number [cmdb_serial_number] table are populated by collected data.

Attribute label	Attribute name
Serial Number	serial_number
Serial Number Type	serial_number_type
Valid	valid

Relationships created for Serial Number

Parent class	Relationship type	Child class
Serial Number [cmdb_serial_number]	Reference	Server [cmdb_ci_server]
Serial Number [cmdb_serial_number]	Reference	Windows Server [cmdb_ci_win_server]
Serial Number [cmdb_serial_number]	Reference	Linux Server [cmdb_ci_linux_server]

Windows Server [cmdb_ci_win_server]

The following attributes in the Windows Server [cmdb_ci_win_server] table are populated by collected data.

Attribute label	Attribute name
Name	name
Object ID	object_id
Serial number	serial_number
CPU core count	cpu_core_count
Disk space (GB)	disk_space
Is Virtual	virtual
Operating System	os
OS Version	os_version
RAM (MB)	ram
Status	install_status

Relationships created for Windows Server

Parent class	Relationship type	Child class
Windows Server [cmdb_ci_win_server]	Virtualized by::Virtualizes	Virtual Machine Instance [cmdb_ci_vm_instance]

Server [cmdb_ci_server]

The following attributes in the Server [cmdb_ci_server] table are populated by collected data.

Attribute label	Attribute name
Name	name
Object ID	object_id
Serial number	serial_number
CPU core count	cpu_core_count
Disk space (GB)	disk_space
Is Virtual	virtual
Operating System	os
OS Version	os_version
RAM (MB)	ram
Status	install_status

Relationships created for Server

Parent class	Relationship type	Child class
Server [cmdb_ci_server]	Contains::Contained by	Storage Volume [cmdb_ci_storage_volume]
Server [cmdb_ci_server]	Owns::Owned by	Cloud Mgmt Network Interface [cmdb_ci_nic]
Server [cmdb_ci_server]	Virtualized by::Virtualizes	Virtual Machine Instance [cmdb_ci_vm_instance]
Server [cmdb_ci_server]	Reference	Software Installation [cmdb_sam_sw_install]

Software [cmdb_ci_spkg]

The following attributes in the Software [cmdb_ci_spkg] table are populated by collected data.

Attribute label	Attribute name
Key	key
Name	name
Version	version

Relationship created for Software

Parent class	Relationship type	Child class
Software [cmdb_ci_spkg]	Reference	Software Instance [cmdb_software_instance]

Software Instance [cmdb_software_instance]

The following attributes in the Software Instance [cmdb_software_instance] table are populated by collected data.

Attribute label	Attribute name
Installed on	installed_on
Name	name

Relationship created for Software Instance

Parent class	Relationship type	Child class
Software Instance [cmdb_software_instance]	Reference	Server [cmdb_ci_server]

Storage Volume [cmdb_ci_storage_volume]

The following attributes in the Storage Volume [cmdb_ci_storage_volume] table are populated by collected data.

Attribute label	Attribute name
Object ID	object_id
Volume ID	volume_id
Name	name
Size	size
Size bytes	size_bytes
State	state
Status	install_status

Relationships created for Storage Volume

Parent class	Relationship type	Child class
Storage Volume [cmdb_ci_storage_volume]	Hosted on::Hosts	Azure Datacenter [cmdb_ci_azure_data_center]
Storage Volume [cmdb_ci_storage_volume]	Reference	Key Value [cmdb_key_value]

Virtual Machine Instance [cmdb_ci_vm_instance]

The following attributes in the Virtual Machine Instance [cmdb_ci_vm_instance] table are populated by collected data.

Attribute label	Attribute name
Name	name
Object ID	object_id
State	state
Status	install_status
VM Instance ID	vm_inst_id

Relationships created for Virtual Machine Instance

Parent class	Relationship type	Child class
Virtual Machine Instance [cmdb_ci_vm_instance]	Hosted on::Hosts	Azure Datacenter [cmdb_ci_azure_data_center]
Virtual Machine Instance [cmdb_ci_vm_instance]	Provisioned From::Provisioned	Image [cmdb_ci_os_template]

Parent class	Relationship type	Child class
Virtual Machine Instance [cmdb_ci_vm_instance]	Provisioned From::Provisioned	Hardware Type [cmdb_ci_compute_template]
Virtual Machine Instance [cmdb_ci_vm_instance]	Contains::Contained by	Cloud Mgmt Network Interface [cmdb_ci_nic]
Virtual Machine Instance [cmdb_ci_vm_instance]	Uses:Used by	Storage Volume [cmdb_ci_storage_volume]
Virtual Machine Instance [cmdb_ci_vm_instance]	Reference	Key Value [cmdb_key_value]

Configure an Azure policy to use a single workspace for pulling the software data of multiple virtual machines (VMs) in different subscriptions.

Before you begin

- Ensure that you've configured the Service Graph Connector for Microsoft Azure. For more information, see [Configure guided setup](#).
- Ensure that you have the Resource Policy Contributor and User Access Administrator roles. For more information, see, [What is Azure Policy?](#) in the Azure documentation site.

Role required: Resource Policy Contributor and User Access Administrator of the Azure directory

About this task

Workspaces are associated with single subscriptions only. So, rather than configuring a subscription every time for pulling software data using

the connector, you can add VMs from various subscriptions to a single workspace by using the following Azure policies:

- For Windows-based VMs:
LogAnalyticsExtension_Windows_VM_Deploy.json
- For Linux-based VMs: LogAnalyticsExtension_Linux_VM_Deploy.json

For more information, see [Azure policy definitions](#) on the GitHub repository.

Note: This configuration is optional and is done after you've configured the Service Graph Connector for Microsoft Azure.

Procedure

1. Sign in to the [Azure portal](#).
2. Navigate to **Subscriptions** and then select **Manage policies**.
3. From the left menu, select **Settings > Policies**.
4. On the Policy page, select **Assign policy**.
5. In the **Policy definition** field of the Basics tab, select the ellipsis icon (...).
6. In the Available Definitions side panel, search for the policy based on your machine type.
 - For Windows-based VMs, select **Deploy - Configure Log Analytics extension to be enabled on Windows virtual machines policy**.
 - For Linux-based VMs, select **Deploy - Configure Log Analytics extension to be enabled on Windows virtual machines policy**.
7. Select **Add**.
8. Select the **Parameters** tab.
9. In the **Log Analytics workspace** field of the **Parameters** tab, select the ellipsis icon (...).

10. From the **Subscription** and **workspaces** lists of the Log Analytics workspace side panel, select a subscription and the workspace to connect the multiple VMs with different subscriptions.
11. Select **Review + create** on the Assign policy page.
12. Review the selected options and then select **Create**.
The assignment of the policy could take some time.

Note: The policy enables the role assignments to the VMs of that subscription by installing the extension in the VMs.
13. Repeat steps 10 to 12 for each subscription to be associated with the workspace.
14. Create a remediation task for the policy to complete the compliance process of the existing VMs associated with the subscriptions.
For more information, see [Create a remediation task](#) in the Azure documentation.
15. Add the VMs to the inventory in the linked automation account.
For more information, see [Enable Change Tracking and Inventory from an Automation account](#) in the Azure documentation.

Service Graph Connector for Observability - AppDynamics (1.2.1)

Use the Service Graph Connector for Observability - AppDynamics to ingest CMDB data from an AppDynamics installation using REST APIs. Push events from AppDynamics into ServiceNow with Event Management.

Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

Supported versions

- Supported versions: AppDynamics version 20.3.

- Supported ServiceNow versions:
 - San Diego
 - Tokyo
 - Utah

Guided Setup

The guided setup for the Service Graph Connector for Observability - AppDynamics provides an organized sequence of tasks to configure the integration on your instance. To access the guided setup, see [Configure guided setup](#).

CMDB Integrations Dashboard

The Integration Commons for CMDB store app provides a dashboard with a central view of the status, processing results, and processing errors of all installed integrations. You can see metrics for all integration runs. You can filter the view to a specific CMDB integration, a specific time duration, or a specific integration run. For more details about monitoring AppDynamics integrations in the CMDB Integrations Dashboard, see [Using the CMDB Integrations Dashboard](#).

Data Mapping

Data from the AppDynamics data sources is mapped and transformed into the ServiceNow CMDB Configuration Item (CI) class definitions using the Robust Transform Engine (RTE). Data is inserted into the ServiceNow CMDB using the Identification and Reconciliation Engine (IRE).

You can use the IntegrationHub ETL app to view the data maps. See [IntegrationHub ETL \(3.2\)](#) for more information.

The AppDynamics data sources include the following:

- SG-AppDynamics Application Services
- SG-AppDynamics Servers and Applications
- SG-AppDynamics Server Tags
- SG-AppDynamics Tier to Tier Relationship

For more information on where data is saved when pulling data from AppDynamics, see [CMDB classes targeted](#).

Set up scheduled import jobs to pull in data from AppDynamics into your CMDB.

Before you begin

Dependencies and requirements:

- The [Integration Commons for CMDB](#) store app, which is automatically installed.
- The [CMDB CI Class Models store app](#) store app, which is automatically installed.
- The ITOM Discovery License plugin (com.snc.item.discovery.license). You must activate this plugin.
- ITOM Licensing plugin (com.snc.item.license). For more information, see [Request Discovery](#).
- The Datastream Action plugin (com.glide.hub.action_type.datastream), which is automatically installed.
- Observability Commons for CMDB (sn_observability), which is only required for event ingestion. This must be installed prior to installing the connector for Event Management to work. For more information, see [Observability Commons for CMDB](#) on the ServiceNow Store.

Note: If you have an earlier version of the Service Graph Connector for Observability - AppDynamics, then do not migrate data from the old connector. You must uninstall the previous version and run the new integration.

Starting with the San Diego release, embedded help content will not be visible on the default Polaris theme in the Next Experience. You must activate the embedded help content by completing the following steps:

1. Select a task section in the guided setup, and then click **Configure**.
2. In the menu bar, click the help icon (?

Role required: admin

Procedure

1. Navigate to **All > Service Graph Connector AppDynamics > Setup**.
2. On the Getting started page, select **Get Started**.
3. Configure the authentication credentials and HTTP connection.
 - a. On the Service Graph Connector for Observability - AppDynamics page, in the Configure the connection section, select the task **Configure Credentials**.
 - b. In the Configure Credentials section, configure your credentials.
 - a. Select **Configure**.
 - b. Update the **Name** and **User name** field.

Note: Server Visibility needs to be active for your AppDynamics account. The AppDynamics user requires the AppDynamics role: Applications and Dashboards Viewer (Default) and Server Monitoring User (Default).
 - c. Click **Update** then **Mark as Complete**.
 - c. In the Configure Connection section, configure the connection.
 - a. Click **Configure**.
 - b. Review the fields and enter the controller base URL into the **Host** field.

HTTP(s) Connection form

Field	Description
Name	Name of the connection.
Host	Target host value used by the connection. The Connection URL will automatically fill in the hostname.

Field	Description
Credential	Credential value that is used by this connection.
Connection alias	Connection value that is used to refer to the connection.
URL builder	URL builder that is used to build the connection URL.
Connection URL	Connection URL for the connection. You can either manually enter a URL, or use the URL builder to build the connection string.
Mutual authentication	Option to set the connection with mutual authentication.
Protocol	Underlying protocol used by the connection. Note: Update the Protocol field if you are using anything other than https.
Active	Option to activate the HTTP connection.
Domain	Domain that contains the connection.
Override default port	Target value port that is used by the connection.
Base path	Base path for HTTPS connection.

c. Click **Update** then **Mark as Complete**.

- d. In the Validate data sources section, validate the data sources by selecting **Configure**.

- a. Review the fields on the form.

Data Source form

Field	Description
Name	Unique name of this data source.
Import set table label	Label of the import set table that this data source will produce.
Import set table name	Name of the table that will be created for this data source.
Type	Data storage type of the data to be imported.
Data in single column	Option to set the data in a single column.
Application	Application that contains this record.
Data Loader	Script that loads data in the import set table.

- b. To test the connection, click the **Test Load 20 Records** related link.

When the test is finished, select **Mark as Complete**.

Testing the connection may take a few moments. The page is refreshed to show the test results.

Note: The connection is successful if the **HTTP Status** is **200**. If there is anything displayed in the **Error Code** and **Error Message** fields, then the connection failed and further troubleshooting is required.

- e. In the Push HTTP Request Template section, push the HTTP Request by selecting **Configure**.

- a. Under the Related Links section, click **Push HTTP Request Template**.

- b. Click **Mark as Complete**.

After you push the HTTP request template, multiple API calls are executed to start the event ingestion service in AppDynamics. For more information, see the [Service Graph Connector for Observability AppDynamics](#) article on the ServiceNow Community site.

4. Configure duplicate detection rules.

- a. On the left side bar, select the Configure duplicate detection rules icon ().

- b. On the Service Graph Connector for Observability - AppDynamics page, in the Duplicate detection rules section, select the task **Configure duplicate detection rules**.

- c. On the next page, in the Configure duplicate detection rules section, click **Configure**.

- d. On the CMDB Duplicate Row Rules form, select the rule that you want to activate and update the Active column value to **true**.

Note: To remove fields from being evaluated, add the field names in the Ignore Fields column. To ignore multiple fields, separate the fields with a comma in a separated list.

When you're finished, click **Mark as Complete**.

5. Configure advanced settings.

- a. On the left side bar, click the advanced icon ().

- b. On the Service Graph Connector for Observability - AppDynamics page, in the Advanced section, select the task **Advanced Settings**.
- c. In the Advanced Settings section, click **Configure**.
 - a. Review the set of advanced properties.

Advanced properties

Advanced property	Description
Toggle to populate relationships between tiers	The relationships between tiers will be imported in cmdb_rel_ci.
Toggle to import business transactions from AppDynamics	The business transactions will be imported into cmdb_ci_service_calculated .
Toggle to populate tags for imported servers	The server tags will be imported into cmdb_key_value.
Toggle to import node data from AppDynamics and map to the cmdb_ci_appl hierarchy	The nodes will be imported into cmdb_ci_appl table hierarchy.

The performance impact changes based on the selected advanced setting.

- b. Select the **Yes** check box to activate each property, as needed.
 - c. Click **Save** then **Mark as Complete**.
6. Set up the scheduled import jobs.
- a. On the left side bar, select the scheduled import jobs icon ().

- b. On the Service Graph Connector for Observability - AppDynamics page, under the Set up scheduled import jobs section, select the **Configure the scheduled job** task.
- c. In the Configure the scheduled job section, select **Configure**.
- d. Review the fields.

Scheduled Data Import form

Field	Description
Name	Name of the scheduled job.
Data source	Data source record that defines the data to import.
Run as	Option to run the scheduled job with the credentials of the specified user.
Active	Option to activate the scheduled job. Select this option.
Concurrent Import	Function that loads the data from multiple import sets. The function then processes and transforms the data concurrently.
Partition Method	Partition method for the concurrent import set.
Partition Size	Import set size for early scheduling.
Execute pre-import script	Option to specify a script to run before the import is performed.

Field	Description
Execute post-import script	Option to specify a script to run after the import is performed.
Application	Application that contains this scheduled job.
Run	Frequency of running the import. Set this value to how frequent you want to pull your data.
Conditional	Conditions under which this job is executed.

- e. Select the imports that you want to run, and click **Execute Now** then **Mark as Complete**.

When you complete the guided setup, you can configure the integration to periodically pull data from AppDynamics. The data is saved in tables that extend from the Configuration item [cmdb_ci] table.

The following attributes in the AppDynamics Extension [sn_sg_appd_extension] table are populated by collected data:

Attribute label	Attribute name
AppDynamics ID	appdynamics_id
Controller Name	controller_name
Agent Type	agent_type
Type	type

The following attributes in the Application [cmdb_ci_appl] table are populated by collected data:

Attribute label	Attribute name
Class	sys_class_name
Name	name
Running process command	running_process_command

Relationships created for Application

Parent class	Relationship type	Child class
Application [cmdb_ci_appl]	Runs on::Runs	Server [cmdb_ci_server]
Application [cmdb_ci_appl]	Reference	AppDynamics Extension [sn_sg_appd_extension]

The following attributes in the Calculated Application Service [cmdb_ci_service_calculated] table are populated by collected data:

Attribute label	Attribute name
Name	name
Hide from dashboard	hide_from_dashboard
Metadata	metadata
Operational status	operational_status
Service Populator Status	populator_status
Service Populator	service_populator
Service Type	type
Short Description	short_description

Relationships created for Calculated Application Service

Parent class	Relationship type	Child class
Calculated Application Service [cmdb_ci_service_calculated]	Depends on::Used by	Application [cmdb_ci_appl]
Calculated Application Service [cmdb_ci_service_calculated]	Depends on::Used by	Server [cmdb_ci_server]
Calculated Application Service [cmdb_ci_service_calculated]	Depends on::Used by	Calculated Application Service [cmdb_ci_service_calculated]

The following attributes in the IP Address [cmdb_ci_ip_address] table are populated by collected data:

Attribute label	Attribute name
IP Address	ip_address
Nic	nic
IP version	ip_version
Name	name

Relationship created for IP Address

Parent class	Relationship type	Child class
IP Address [cmdb_ci_ip_address]	Reference	Network Adapter [cmdb_ci_network_adapter]

The following attributes in the Key Value [cmdb_key_value] table are populated by collected data:

Attribute label	Attribute name
Key	key
Value	value

The following attributes in the Network Adapter [cmdb_ci_network_adapter] table are populated by collected data:

Attribute label	Attribute name
MAC Address	mac_address
Name	name
Discovery source	discovery_source

Relationship created for Network Adapter

Parent class	Relationship type	Child class
Network Adapter [cmdb_ci_network_adapter]	Reference	Server [cmdb_ci_server]

The following attributes in the Server [cmdb_ci_server] table are populated by collected data:

Attribute label	Attribute name
Name	name

Relationships created for Server

Parent class	Relationship type	Child class
Server [cmdb_ci_server]	Owns::Owned by	IP Address [cmdb_ci_ip_address]

Parent class	Relationship type	Child class
Server [cmdb_ci_server]	Owns::Owned by	Network Adapter [cmdb_ci_network_adapter]
Server [cmdb_ci_server]	Reference	Key Value [cmdb_key_value]

Service Graph Connector for Observability - Datadog (1.2.1)

Use the Service Graph Connector for Observability - Datadog to ingest CMDB data from a Datadog installation using REST APIs. Push events from Datadog into ServiceNow with Event Management.

Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

Supported versions

Supported ServiceNow versions:

- San Diego
- Tokyo
- Utah

Guided Setup

The guided setup for the Service Graph Connector for Observability - Datadog provides an organized sequence of tasks to configure the integration on your instance. To access the guided setup, see [Configure guided setup](#).

CMDB Integrations Dashboard

The Integration Commons for CMDB store app provides a dashboard with a central view of the status, processing results, and processing errors of all installed integrations. You can see metrics for all integration runs. You can filter the view to a specific CMDB integration, a specific time duration, or a specific integration run. For more details about monitoring Observability Datadog integrations in the CMDB Integrations Dashboard, see [Using the CMDB Integrations Dashboard](#).

Data Mapping

Data from the Datadog data sources is mapped and transformed into the ServiceNow CMDB Configuration Item (CI) class definitions using the Robust Transform Engine (RTE). Data is inserted into the ServiceNow CMDB using the Identification and Reconciliation Engine (IRE).

You can use the IntegrationHub ETL app to view the data maps. See [IntegrationHub ETL \(3.2\)](#) for more information.

The Datadog data source includes SGO-Datadog Hosts.

For more information on where data is saved when pulling data from Datadog, see [CMDB classes targeted](#).

When you complete the guided setup, the data from Datadog is automatically loaded into staging tables and then inserted into the following target tables:

- Cloud DataBase [cmdb_ci_cloud_database]
- Cloud Load Balancer [cmdb_ci_cloud_load_balancer]
- Cloud Service Account [cmdb_ci_cloud_service_account]
- IP address [cmdb_ci_ip_address]
- Logical Datacenter [cmdb_ci_logical_datacenter]
- Network Adapter [cmdb_ci_network_adapter]
- Server [cmdb_ci_server]

Pull in data from Datadog into your CMDB.

Before you begin

Dependencies and requirements:

- The [Integration Commons for CMDB](#) store app, which is automatically installed.
- The [CMDB CI Class Models store app](#) store app, which is automatically installed.
- The ITOM Discovery License plugin (com.snc.item.discovery.license). You must activate this plugin.
- ITOM Licensing plugin (com.snc.item.license). For more information, see [Request Discovery](#).
- The Datastream Action plugin (com.glide.hub.action_type.datastream), which is automatically installed.
- Observability Commons for CMDB (sn_observability), which is only required for event ingestion. This must be installed prior to installing the connector for Event Management to work. For more information, see [Observability Commons for CMDB](#) on the ServiceNow Store.

Note: If you have an earlier version of the Service Graph Connector for Observability - Datadog, then don't migrate data from the old connector. You must uninstall the previous version and run the new integration.

Starting with the San Diego release, embedded help content won't be visible on the default Polaris theme in the Next Experience. You must activate the embedded help content by completing the following steps:

1. Select a task section in the guided setup, and then click **Configure**.



2. In the menu bar, click the help icon ().

Role required: admin

Procedure

1. Navigate to **All > Service Graph Connectors > Observability Datadog > Setup**.
2. On the Getting started page, select **Get Started**.
3. Configure the authentication credentials and the host name to send requests to the Datadog API.
 - a. Configure your Datadog credentials.
 - a. In the Configure the Connection section of the Datadog Integration with CMDB page, select **Get Started**.
 - b. For the Configure the Credentials task, select **Configure**.
 - c. In the **Name** field, enter a name for the authentication. For example, **Datadog credentials**.
 - d. In the **API Key** field, enter the Datadog API token.
 - e. In the **Authentication Key** field, enter the authentication key used for connecting to the Datadog API.
 - f. Click **Update**.
 - g. Set the Configure the Credentials task to complete by clicking **Mark as Complete**.
 - b. Configure the Datadog connection settings.
 - a. In the Configure the Connection section of the Datadog Integration with CMDB page, select **Continue**.
 - b. For the Configure the Connection task, select **Configure**.
 - c. Review the fields and in the **Host** field, enter the Datadog base URL or IP address.
 - d. Update the **Protocol** field if you're using anything other than https.
 - e. Leave the **Base path** field as is.

f. Click **Update**.

g. Set the Configure the Connection task to complete by clicking **Mark as Complete**.

For more information about the Datadog API, see the [Datadog Developer documentation](#).

4. Configure a user account for the Datadog and CMDB integration.

- a. Create the Datadog CMDB integration user account by completing the User form.
 - a. In the Datadog CMDB integration user account section of the Datadog Integration with CMDB page, select **Get Started**.
 - b. For the Create a Datadog CMDB integration user task, select **Configure**.
 - c. Review the fields on the User form, enter the Datadog CMDB integration user account details, and set a password for the user account.
 - d. Click **Update**.
 - e. Set the Create a Datadog CMDB integration user task to complete by clicking **Mark as Complete**.
- b. Assign the `cmdb_import_api_admin` role to the Datadog CMDB integration user.
 - a. In the Datadog CMDB integration user account section of the Datadog Integration with CMDB page, select **Continue**.
 - b. For the Assign role to the Datadog CMDB user task, select **Configure**.
 - c. In the User ID column of the Users list, select the Datadog CMDB integration user ID you created in the earlier step 4.a.
 - d. In the Roles related list, click **Edit**.
 - e. On the Edit Members form, move the `cmdb_import_api_admin` role from the available roles in the **Collection** column to the **Roles List** column.

- f. Click **Save**.
 - g. Click **Update**.
 - h. Set the Assign role to the Datadog CMDB user task to complete by clicking **Mark as Complete**.
5. Configure the ServiceNow tile on Datadog and enable the Datadog CMDB integration.
 - a. Configure the ServiceNow tile on Datadog.
 - a. Log in to your Datadog account.
 - b. Select the Integrations tab.
 - c. Search for and select the ServiceNow tile.
 - d. Add the ServiceNow instance name in the format https://<instance name>.service-now.com/.
 - e. Add the name and password for the Datadog CMDB integration user account and click **Submit**.

For more information, see the [Datadog documentation](#).

Note: After you have configured the ServiceNow tile on Datadog, in the guided setup for Service Graph Connector for Observability - Datadog, go to the Configure Datadog tile and enable integration section of the Datadog Integration with CMDB page and set the Configure the ServiceNow tile in Datadog task to complete by clicking **Mark as Complete**.

- b. Enable the Datadog CMDB integration on your ServiceNow instance.
 - a. In the Configure Datadog tile and enable integration section of the Datadog Integration with CMDB page, select **Continue**.
 - b. For the Enable the integration on ServiceNow task, select **Configure**.

- c. On the System Property form, fill in the fields to create the sn_datadog_integra.datadog_enabled system property and set its value to true.

For more information, see [Add a system property](#).

- d. Click **Submit**.
- e. Set the Enable the integration on ServiceNow task to complete by clicking **Mark as Complete**.

6. Configure the webhook and monitors for Observability Datadog.

- a. In the Configure Observability section of the Datadog Integration with CMDB page, select **Get started**.
- b. For the Configure the Webhooks and Monitors task, select **Configure**.
- c. In the Datadog Webhooks list, click **New** to add a Datadog webhook.
- d. Fill in the fields.

New record form

Field	Description
Name	Name of the Datadog webhook.
Connection Alias	Search for and select the connection and credential alias you created in step 3.

- e. Click **Submit**.
- f. Populate the Datadog Webhooks list by clicking the **Synchronize Monitors** related link on the Datadog Webhook page.

Note: In the Datadog Webhooks list, don't click **New** to add a monitor as the button is not used in this scenario.

- g. In the **Name** column of the Datadog Webhooks list, click the link to a monitor.

- h. On the Datadog Monitors page, select the **Webhook Active** check box and then click **Update**.
- i. Repeat steps [6.g](#) and [6.h](#) for all the monitors in the Datadog Webhooks list.
- j. On the Datadog Webhook page, click **Update** to save your changes.
- k. Set the Configure the Webhooks and Monitors task to complete by clicking **Mark as Complete**.

When you complete the guided setup, the data is automatically pulled from Datadog. The data is saved in target tables.

The following attributes in the Server [cmdb_ci_server] table are populated by collected data:

Attribute label	Attribute name
Class	sys_class_name
CPU core thread	cpu_core_thread
CPU count	cpu_count
CPU speed (MHz)	cpu_speed
CPU type	cpu_type
Disk space (GB)	disk_space
DNS Domain	dns_domain
Fully qualified domain name	fqdn
Operating System	os
RAM (MB)	ram

Relationships created for Server

Parent class	Relationship type	Child class
Server [cmdb_ci_server]	Owns::Owned by	Network Adapter [cmdb_ci_network_adapter]
Server [cmdb_ci_server]	Owns::Owned by	IP Address [cmdb_ci_ip_address]

The following attributes in the Cloud Service Account [cmdb_ci_cloud_service_account] table are populated by collected data:

Attribute label	Attribute name
Account Id	account_id
Name	name
Object ID	object_id
Datacenter Type	datacenter_type

The following attributes in the Cloud Load Balancer [cmdb_ci_cloud_load_balancer] table are populated by collected data:

Attribute label	Attribute name
Name	name
Object ID	object_id

Relationship created for Cloud Load Balancer

Parent class	Relationship type	Child class
Cloud Load Balancer [cmdb_ci_cloud_load_balancer]	Hosted on::Hosts	Logical Datacenter [cmdb_ci_logical_datacenter]

The following attributes in the IP Address [cmdb_ci_ip_address] table are populated by collected data:

Attribute label	Attribute name
IP Address	ip_address
Nic	nic
Name	name

Relationship created for IP Address

Parent class	Relationship type	Child class
IP Address [cmdb_ci_ip_address]	Reference	Network Adapter [cmdb_ci_network_adapter]

The following attributes in the Cloud DataBase [cmdb_ci_cloud_database] table are populated by collected data:

Attribute label	Attribute name
Object ID	object_id
Fully qualified domain name	fqdn
Type	type
Version	version

Relationship created for Cloud DataBase

Parent class	Relationship type	Child class
Cloud DataBase [cmdb_ci_cloud_data]	Hosted on::Hosts base	Logical Datacenter [cmdb_ci_logical_dat acenter]

The following attributes in the Network Adapter [cmdb_ci_network_adapter] table are populated by collected data:

Attribute label	Attribute name
Configuration Item	cmdb_ci
MAC Address	mac_address
Name	name

Relationship created for Network Adapter

Parent class	Relationship type	Child class
Network Adapter [cmdb_ci_network_adapter]	Reference	Server [cmdb_ci_server]

The following attributes in the Logical Datacenter [cmdb_ci_logical_datacenter] table are populated by collected data:

Attribute label	Attribute name
Name	name
Object ID	object_id
Region	region

Relationship created for Logical Datacenter

Parent class	Relationship type	Child class
Logical Datacenter [cmdb_ci_logical_datacenter]	Hosted on::Hosts	Cloud Service Account [cmdb_ci_cloud_service_account]

Service Graph Connector for Observability - Dynatrace (1.8.0)

Use the Service Graph Connector for Observability - Dynatrace to ingest CI data, events, metrics, and logs from Dynatrace into your ServiceNow instance.

Note: The Service Graph Connector for Observability - Dynatrace provided by ServiceNow is different from the Service Graph Connector for Dynatrace provided by Dynatrace.

Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

Supported versions

- Supported versions:
 - Dynatrace Release 177
 - Dynatrace Release 224: This version is optional for pushing configuration to Dynatrace for Event Management.
- Supported ServiceNow versions:
 - San Diego
 - Tokyo
 - Utah
 - Vancouver

Guided Setup

The guided setup for the Service Graph Connector for Observability - Dynatrace provides an organized sequence of tasks to configure the integration on your instance. To access the guided setup, see [Configure guided setup](#).

CMDB Integrations Dashboard

The Integration Commons for CMDB store app provides a dashboard with a central view of the status, processing results, and processing errors of all installed integrations. You can see metrics for all integration runs. You can filter the view to a specific CMDB integration, a specific time duration, or a specific integration run. For more details about monitoring Dynatrace integrations in the CMDB Integrations Dashboard, see [Using the CMDB Integrations Dashboard](#).

Data mapping

Data from the Dynatrace data sources is mapped and transformed into the ServiceNow CMDB Configuration Item (CI) class definitions using the Robust Transform Engine (RTE). Data is inserted into the ServiceNow CMDB using the Identification and Reconciliation Engine (IRE).

You can use the IntegrationHub ETL app to view the data maps. See [IntegrationHub ETL \(3.2\)](#) for more information.

The Dynatrace data sources include the following:

- SGO-Dynatrace Applications
- SGO-Dynatrace Application Relationships
- SGO-Dynatrace Hosts
- SGO-Dynatrace Processes

Note: Beginning with the 1.8.0 version of the Service Graph Connector for Observability - Dynatrace, the SG-Dynatrace Processes Transform Map includes the onStart transform script that transforms all the processes fetched from Dynatrace using the `DynatraceADMprocessor` class. The earlier robust transformer is no longer used.

- SGO-Dynatrace Process Groups
- SGO-Dynatrace Services

For more information on where data is saved when pulling data from Dynatrace, see [CMDB classes targeted](#).

Use the Service Graph Connector for Observability - Dynatrace to ingest Configuration Management Database (CMDB) data from Dynatrace using REST APIs. This connector is the second generation of the Service Graph Connector for Observability - Dynatrace application developed by ServiceNow.

Before you begin

Dependencies and requirements:

- The [Integration Commons for CMDB](#) store app, which is automatically installed.
- The [CMDB CI Class Models](#) store app store app, which is automatically installed.
- The ITOM Discovery License plugin (com.snc.itom.discovery.license). You must activate this plugin.
- ITOM Licensing plugin (com.snc.itom.license). For more information, see [Request Discovery](#).
- The Datastream Action plugin (com.glide.hub.action_type.datastream), which is automatically installed.
- Observability Commons for CMDB (sn_observability), which is only required for event ingestion and will need to be installed prior to installing the connector for Event Management to work. For more information, see [Observability Commons for CMDB](#) on the ServiceNow Store.

Starting with the San Diego release, embedded help content will not be visible on the default Polaris theme in the Next Experience. You must activate the embedded help content by completing the following steps:

1. Select a task section in the guided setup, and then click **Configure**.
2. In the menu bar, click the help icon (?

Role required: admin

Procedure

1. Navigate to **All > Service Graph Connectors > Dynatrace Observability > Setup**.

2. On the Getting started page, select **Get Started**.

3. Migrate from the previous Dynatrace installation.

Note: If you do not have a previous version of Dynatrace installed, you can skip this step.

a. On the Setup page, in the Migrate from previous Dynatrace installation section, select the task **Disable Previous Scheduled Jobs**.

b. Disable the previous scheduled jobs.

a. In the Disable Previous Scheduled jobs section, click **Configure**.

b. On the Scheduled Jobs list, set the **Active** field to **false**.

If this list is empty, then there are no scheduled jobs that need to be deactivated.

c. Click **Mark as Complete**.

c. Copy the migration script.

a. In the Migrate Data - Copy Migration Script section, click **Configure**.

b. Change the current scope to **Global**.

c. Click the menu icon (≡).

d. Click **Insert and Stay**.

e. Click **Mark as Complete**.

d. Execute the script.

a. In the Migrate Data - Execute Script section, click **Configure**.

- b. Click the record of the script that you copied.
 - c. Click **Execute Now**.
4. Set impact values for clusters and have the connector get access to the SNC.ImpactManager API.

Note: This step appears when Observability Commons is installed.

 - a. On the Setup page, under the Enable Access To SNC.Impact Manager, select the **Copy Script to Global Scope** task.
 - b. In the Copy Script to Global Scope section, copy the script.
 - a. Click **Configure**.
 - b. Switch to the global scope.
 - c. Click the Additional actions icon (≡).
 - d. Select **Insert and Stay**.
 - e. Click **Mark as Complete**.
 - c. In the Verify script is copied properly section, verify the script.
 - a. Click **Configure**.
 - b. Select the EvtMgmtImpactManagerMediator script and verify it was copied to the global scope.
 - c. In the **Accessible from** field, ensure it is set to **All application scopes**.
 - d. Click **Mark as Complete**.
5. Configure the basic setup.
 - a. On the Setup page, under the Basic section, select the **Configure Auth Token for Dynatrace** task.
 - b. On the next page, in the Configure Auth Token for Dynatrace section, configure the authentication token.
 - a. Click **Configure**.

- b. In the **API Key** field, enter `api-token <your api token>`.

For example, `api-token mytokenid`.

- c. Click **Mark as Complete**.

- c. In the Configure HTTP Connection for Dynatrace section, configure the HTTP connection.

- a. Click **Configure**.

- b. Update the **Host** field with a fully qualified hostname for your Dynatrace instance.

For example, `abc123.live.dynatrace.com`.

The hostname will be automatically filled in the **Connection URL** field.

- c. Enable the use of a MID Server, select the **Use MID Server** check box.

Note: The HTTP connection will be pre-configured to use the API key that was configured during the previous setup task.

- d. Click **Mark as Complete**.

- d. In the Test Connection section, to test the connection, select **Configure**.

- a. In the Test Connection section, click **Configure**.

- b. To test the connection configuration, select **Test Connection**.

Note: If the test connection fails, there is an error in the connection that you must fix.

- c. When you're finished, click **Mark as Complete**.

- e. In the Create Default Notification Payload Template section, select **Configure**.

Note: You need an access token with the following scopes:

- Read configuration (ReadConfig)
 - Write configuration (WriteConfig)
 - Read logs (LogExport)
 - Read metrics (metrics.read)
- a. Update the name of the payload template, if needed.
 - b. Click **Problem Notification Setup**.
 - c. When you're finished, click **Mark as Complete**.
- f. (Optional) If you want to support multi-instance, in the Upgrade Source Native Keys section, click **Configure**.
- a. Switch to the global scope.
 - b. Enter the following script.

```
var gr = new GlideRecord("sys_object_source");
gr.addQuery("name", "SGO-Dynatrace");
var grOR = gr.addQuery("id", "STARTSWITH", "HOST-");
grOR.addOrCondition("id", "STARTSWITH", "PROCESSES_GROUP_INSTANCE-");
grOR.addOrCondition("id", "STARTSWITH", "PROCESSES_GROUP-");
grOR.addOrCondition("id", "STARTSWITH", "SERVICE-");
grOR.addOrCondition("id", "STARTSWITH", "APPLICATION-");
gr.query();
while (gr.next()) {
    gr.setValue("id", "f379137e075820107add6a77c4a93538" || gr.getValue("id"));
    gr.update();
}
```

- c. Select **Run Script**.

- d. From the Scope menu, select **Service Graph Connector for Observability Dynatrace** then **Mark as Complete**.
6. Set up the additional configurations.
 - a. On the Guided setup page, under the Advanced section, select the **Advanced settings** task.
 - b. In the Advanced Settings section, select **Configure** and review or modify the existing settings for a custom configuration.
You can configure the following settings:
 - Review the page size used in REST API requests to fetch Dynatrace entities.
 - Define number of days a configuration item (CI) can be inactive before it is ignored.
 - Enter the percentage of an application cluster's nodes that need to be in a state to raise that state to its parent in the service map.

For example, you can define the percentage of nodes that need to go critical for the parent of a cluster to be in a critical state. If there are 10 nodes in a cluster, setting the property value to 70 would require at least 7 out of the 10 nodes in the cluster to go into a critical state to reflect up to the parent service of the cluster.

 - Enable ingesting events that do not have a matching CI in the CMDB.
 - Enable populating the Application (cmdb_ci_appl) CIs from Dynatrace (Dynatrace processes) during scheduled imports.
 - c. Click **Save**.
 - d. Select **Mark as Complete** for the Advanced Settings task.
 - e. Configure connection properties for the Dynatrace connection.
 - a. In the Configure Instance Settings section, select **Configure**.
 - b. In the Service Graph Connection Properties related list, configure the properties of the connection record.

Dynatrace connection properties

Property	Description
managementZoneNames	Enter the name of the management zone to fetch from your Dynatrace environment. For multiple entries, separate the zone names with commas.
tags	Enter the name of the tags to fetch from your Dynatrace environment. For multiple entries, separate the zone names with commas.
serviceTypes	Enter the name of the management zone to fetch from your Dynatrace environment. For multiple entries, separate the zone names with commas.
managementZonelds	Enter the name of the management zone to fetch from your Dynatrace environment. For multiple entries, separate the zone names with commas.

- c. Click **Update**.
- d. Complete the Configure Instance Settings task by clicking **Mark as Complete**.

Note: Beginning with the 1.8.0 version of the Service Graph Connector for Observability - Dynatrace, the Dynatrace Connections [sn_dynatrace_integ_connections] table is being migrated to the Service Graph Connections [sn_cmdb_int_util_service_graph_connection] table. The Migrating Dynatrace Connections fix script that is available by default migrates the details of any existing connections, single instance or multi-instance, including the connection name, alias, and status to the Service Graph Connections [sn_cmdb_int_util_service_graph_connection] table. The fix script also migrates any connection properties, data sources, and scheduled data imports to the corresponding related tables.

f. Configure the notification settings.

The setup enables pulling ITOM events from Dynatrace into the ServiceNow instance.

- a. In the Configure Problem Notification section, select **Configure**.
- b. To push the configuration to Dynatrace, select **Problem Notification Setup**.
- c. To receive the configuration from Dynatrace, select **Fetch Notification Setup**.
- d. Complete the Configure Problem Notification task by clicking **Mark as Complete**.

7. Clean up records from the previous instance.

Note: If you are not migrating from the previous Dynatrace version, you can skip this step.

- a. On the Setup page, under the Clean Up Records From Previous Integration section, select the **Execute the New Integration** task.
- b. On the next page, in the Execute the New Integration section, select **Configure** and select **Execute Now**.
- c. Select **Mark as Complete**.

- d. In the Verify Integration Execution Has Completed section, verify the integration execution.
 - a. Click **Configure**.
 - b. In the **State** field, wait for the field to change to **Complete** or **Complete with errors**.

It is normal to have errors during the migration from an older version.
 - c. If you need to refresh the list, right-click the header and select **Refresh List**.
 - d. Click **Mark as Complete**.
 - e. In the Delete Application Services Left Over From Previous Version section, select **Configure** to delete any application services that you no longer need from the previous integration.
 - f. Click **Mark as Complete**.
 - g. In the Cleanup Identification Remnants - Copy Script section, copy the script.
 - a. Click **Configure**.
 - b. Change the current scope to **Global**.
 - c. Click the menu icon ().
 - d. Click **Insert and Stay**.
 - e. Click **Mark as Complete**.
 - h. In the Cleanup Identification Remnants - Execute Script section, execute the script.
 - a. Click **Configure**.
 - b. Select the record of the script you copied.
 - c. Click **Execute Now**.
8. (Optional) Add multiple instances.

Note: If you do not need to add multiple instances, you can skip this step.

- a. On the left side bar, select the Add Multiple Instances icon ().
 - b. On the Service Graph Connector for Observability Dynatrace Setup page, under the Add Multiple Instances section, select the **Update Data Source Access** task.
 - c. On the next page, in the Update Data Source Access section, select **Configure**.
 - d. Select the Data Source [sys_data_source] table.
 - e. To edit the record, select **Global** from the Scope menu.
 - f. Under the **Application Access** tab, select the **Can create**, **Can update**, and **Can delete** check boxes.
 - g. Save the record.
 - h. From the Scope menu, select **Service Graph Connector for Observability Dynatrace**.
 - i. In the Help task bar, click **Mark as Complete**.
 - j. Repeat these steps in the Update Scheduled data import access section with the Scheduled data import [scheduled_data_set] table and the Update Value Access section with the Value [sys_variable_table] table.
9. Clear the cache for the new connection.
- a. Select the **Clear Cache for Datasource and Import set** task, then **Configure**.
 - b. Clear the cache by selecting **Global** from the Scope menu.
 - c. Enter the following script.

```
GlideTableManager.invalidateTable("sys_data_source");
GlideCacheManager.flushTable("sys_data_source");
```

```
GlideTableManager.invalidateTable("scheduled_import_set");
GlideCacheManager.flushTable("scheduled_import_set");

GlideTableManager.invalidateTable("sys_variable_value");
GlideCacheManager.flushTable("sys_variable_value");

GlideTableManager.invalidateTable("sys_db_object");
GlideCacheManager.flushTable("sys_db_object");
```

- d. Select **Run Script**.
- e. From the Scope menu, select **Service Graph Connector for Observability Dynatrace**.
- f. Click **Mark as Complete**.

10. Add another connection.

Note: Change the scope to **Service Graph Connector for Observability**, otherwise you will be unable to load the additional connections.

- a. Under the Add Another Connection section, click **Configure**.
- b. In Flow Designer, select **Add Connection**.
- c. On the form, fill in the fields.

Connection form

Field	Description
Connection Name	Display name for the connection.
Connection Hostname	Host name of the Dynatrace instance.

Field	Description
API Key	Dynatrace API Key. Note: The API Key must be prefixed with api-token.

- d. Click **Create Connection**.
- e. Navigate back to the guided setup and click **Mark as Complete**.
- f. If needed, set up the MID Server for the connection you created.
 - a. In the Configure Mid Servers section, click **Configure**.
 - b. Select the name of the connection you created.
 - c. Click the **Use MID server** check box.
 - d. Click **Update**.
 - e. When you're finished with the task, click **Mark as Complete**
- g. Configure the instance settings.
 - a. In the Configure Instances section, click **Configure**.
 - b. Select the name of the connection you want to configure.
 - c. When you're finished, click **Update** then **Mark as Complete**.
- h. Test the new connections.
 - a. In the Test Connections section, click **Configure**.
 - b. Select the name of the connection you want to test.
 - c. To validate the data source configuration, click the **Test Load 20 Records** button.
Note: If the test connection fails, there is an error in the connection that you must fix.
 - d. When you're finished, click **Mark as Complete**.

- i. In the Create Default Notification Payload Templates section, select **Configure**.

Note: You need an access token with the following scopes:

- Read configuration (ReadConfig)
- Write configuration (WriteConfig)
- Read logs (LogExport)
- Read metrics (metrics.read)

- a. Select the connection you want to create a default notification payload template for.
- b. Update the name of the payload template, if needed.
- c. Click **Problem Notification Setup**.
- d. Repeat the steps for each connection.
- e. When you're finished, click **Mark as Complete**.

11. Set up scheduled import jobs.

- a. On the Setup page, under the Set up scheduled import jobs section, select the **Configure the scheduled import jobs** task.
- b. On the next page, select **Configure**.
- c. On the form, review the fields as needed.

Scheduled Data Import form

Field	Description
Name	Name of the scheduled job.
Data source	Data source record that defines the data to import.
Run as	Option to run the scheduled job with the credentials of the specified user.

Field	Description
Active	Option to activate the scheduled job. Select this option.
Concurrent Import	Function that loads the data from multiple import sets. The function then processes and transforms the data concurrently.
Partition Method	Partition method for the concurrent import set.
Partition Size	Import set size for early scheduling.
Execute pre-import script	Option to specify a script to run before the import is performed.
Execute post-import script	Option to specify a script to run after the import is performed.
Application	Application that contains this scheduled job.
Run	Frequency of running the import. Set this value to how frequent you want to pull your data.
Conditional	Conditions under which this job is executed.

- d. Select the imports that you want to run, and click **Execute Now** then **Mark as Complete**.

When you complete the guided setup, you can configure the integration to periodically pull data from Dynatrace. The data is saved in tables that extend from the Configuration item [cmdb_ci] table.

Relationships created for Configuration Item

Parent class	Relationship type	Child class
Configuration Item [cmdb_ci]	Depends on::Used by	Calculated Application Service [cmdb_ci_service_calculated]
Configuration Item [cmdb_ci]	Depends on::Used by	Database Instance [cmdb_ci_db_instance]

The following attributes in the Application [cmdb_ci_appl] table are populated by collected data:

Attribute label	Attribute name
Class	sys_class_name
Name	name
Running process command	running_process_command
Running process key parameters	running_process_key_parameters
Configuration file	config_file
Installation directory	install_directory
Operational status	operational_status
Version	version

Relationships created for Application

Parent class	Relationship type	Child class
Application [cmdb_ci_appl]	Runs on::Runs	Computer [cmdb_ci_computer]
Application [cmdb_ci_appl]	Reference	Key value [cmdb_key_value]

The following attributes in the Calculated Application Service [cmdb_ci_service_calculated] table are populated by collected data:

Attribute label	Attribute name
Name	name
Correlation ID	correlation_id
Hide from dashboard	hide_from_dashboard
Metadata	metadata
Operational status	operational_status
Service Populator Status	populator_status
Service Populator	service_populator
Service Type	type

Relationships created for Calculated Application Service

Parent class	Relationship type	Child class
Calculated Application Service [cmdb_ci_service_calculated]	Depends on::Used by	Configuration Item [cmdb_ci]
Calculated Application Service	Reference	Key value [cmdb_key_value]

Parent class	Relationship type	Child class
[cmdb_ci_service_calculated]		

The following attributes in the Computer [cmdb_ci_computer] table are populated by collected data:

Attribute label	Attribute name
Name	name
Class	sys_class_name
CPU core count	cpu_core_count
DNS Domain	dns_domain
Fully qualified domain name	fqdn
Is Virtual	virtual
Operating System	os
OS Version	os_version
RAM (MB)	ram

Relationships created for Computer

Parent class	Relationship type	Child class
Computer [cmdb_ci_computer]	Owns::Owned by	IP Address [cmdb_ci_ip_address]
Computer [cmdb_ci_computer]	Reference	Key Value [cmdb_key_value]
Computer [cmdb_ci_computer]	Reference	Software Installation [cmdb_sam_sw_install]

The following attributes in the Database Instance [cmdb_ci_db_instance] table are populated by collected data:

Attribute label	Attribute name
Edition	edition
Name	name
TCP port(s)	tcp_port

Relationships created for Database Instance

Parent class	Relationship type	Child class
Database Instance [cmdb_ci_db_instance]	Depends on::Used by	Configuration Item [cmdb_ci]
Database Instance [cmdb_ci_db_instance]	Contains::Contained by	Group [cmdb_ci_group]
Database Instance [cmdb_ci_db_instance]	Depends on::Used by	Application [cmdb_ci_appl]
Database Instance [cmdb_ci_db_instance]	Reference	Key Value [cmdb_key_value]

The following attribute in the Group [cmdb_ci_group] table is populated by collected data:

Attribute label	Attribute name
Name	name

Relationship created for Group

Parent class	Relationship type	Child class
Group [cmdb_ci_group]	Contains::Contained by	Application [cmdb_ci_appl]

The following attributes in the IP Address [cmdb_ci_ip_address] table are populated by collected data:

Attribute label	Attribute name
IP Address	ip_address
IP version	ip_version
Name	name

The following attributes in the Key value [cmdb_key_value] table are populated by collected data:

Attribute label	Attribute name
Key	key
Value	value

The following attributes in the Software [cmdb_ci_spkg] table are populated by collected data:

Attribute label	Attribute name
Key	key
Name	name
Version	version

Relationship created for Software

Parent class	Relationship type	Child class
Software [cmdb_ci_spkg]	Reference	Software Instance [cmdb_software_instance]

The following attributes in the Software Installation [cmdb_sam_sw_install] table are populated by collected data:

Attribute label	Attribute name
Display name	display_name
Version	version

The following attributes in the Software Instance [cmdb_software_instance] table are populated by collected data:

Attribute label	Attribute name
Name	name
Installed on	installed_on

Relationship created for Software Instance

Parent class	Relationship type	Child class
Software Instance [cmdb_software_instance]	Reference	Computer [cmdb_ci_computer]

Set up push notifications of events from Dynatrace into a ServiceNow instance that has the Service Graph Connector for Observability - Dynatrace installed.

Before you begin

If you don't have an alerting profile in Dynatrace, then complete the following steps to create one:

1. Navigate to **Settings > Alerting > Alerting Profiles**.
2. Set up an alerting profile according to your business needs.

Note: For more information about how to set up an alerting profile, see [Alerting profiles](#) on the Dynatrace documentation site.

You must have Observability Commons for CMDB installed.

Role required: admin

Procedure

1. In the Dynatrace instance, navigate to **Settings > Integration > Problem notifications.**
2. Click **Add Notification.**
3. On the form, fill in the fields.

The screenshot shows a configuration page for a 'Custom Integration'. The top section is titled 'Custom Integration' and contains several input fields:

- A large text input field containing the letters 'N', 'o', 't', 'i', 'f', 'i', 'c', 'a', 'f', 'i', 'o', 'n', 'T', 'y', 'p', 'e'.
- A smaller text input field containing the letters 'D', 'i', 's', 'p', 'l', 'y', 'N', 'a', 'm', 'e'.
- An input field labeled 'Name of the notification' with the text 'Name of the notification'.

Meld	
W	
e	
b	
h	
u	se the following URL: <a href="https://<name_of_your_servicenow_instance_name>/api/sn_em_connector/em/inbound_event?source=SGO-Dynatrace">https://<name_of_your_servicenow_instance_name>/api/sn_em_connector/em/inbound_event?source=SGO-Dynatrace
k	
U	
R	
L	
A	
d	
d	
i	
t	
i	
o	
na.	In the ServiceNow instance, set a password for the pre-created Dynatracee API user by doing the following:
I	a. Navigate to System Security > Users .
T	b. Select the Dynatrace API user.
P	c. Set the Password field with a new password.
H	d. Clear the Password needs reset check box.
e	e. Update or save the user.
r	b. Set the Username field to DynatraceAPI .
s	c. Set the Password field to the password that you had created.
C	d. Click Add .
e	
a	
t	
e	
b	
a	

Melde

s
i
c
a
u
t
h
o
r
i
z
a
t
i
o
n

Use the following payload:

```
{  
    "ImpactedEntities": {ImpactedEntities},  
    "ImpactedEntity": "{ImpactedEntity}",  
    "PID": "{PID}",  
    "ProblemDetailsHTML": "{ProblemDetailsHTML}",  
    "ProblemDetailsJSON": {ProblemDetailsJSON},  
    "ProblemDetailsMarkdown": "{ProblemDetailsMarkdown}",  
    "ProblemDetailsText": "{ProblemDetailsText}",  
    "ProblemID": "{ProblemID}",  
    "ProblemImpact": "{ProblemImpact}",  
    "ProblemSeverity": "{ProblemSeverity}",  
    "ProblemTitle": "{ProblemTitle}",
```

Melde

```
"ProblemURL": "{ProblemURL}" ,  
"State": "{State}" ,  
"Tags": "{Tags}"  
}
```

A
l
e
r
t
i
n
Select the alerting profile created in the Before you begin section or another alerting profile.
P
r
o
f
i
l
e

S
e
n
d
i
n
g
T
Click to send a test notification, and verify that the response status is 200.
e
s
t
N
o
t
i
f

Meldei
c
a
t
i
o
n
s

Service Graph Connector for Observability - New Relic (1.2.1)

Use the Service Graph Connector for Observability - New Relic to ingest CMDB data from a New Relic installation using REST APIs. Push events from New Relic into ServiceNow with Event Management.

Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

Supported versions

- New Relic version: Last tested on March 01, 2023
- Supported ServiceNow versions:
 - San Diego
 - Tokyo
 - Utah

Guided Setup

The guided setup for the Service Graph Connector for Observability - New Relic provides an organized sequence of tasks to configure the integration on your instance. To access the guided setup, see [Configure guided setup](#).

CMDB Integrations Dashboard

The Integration Commons for CMDB store app provides a dashboard with a central view of the status, processing results, and processing errors of all installed integrations. You can see metrics for all integration runs. You can filter the view to a specific CMDB integration, a specific time duration, or a specific integration run. For more details about monitoring Observability New Relic integrations in the CMDB Integrations Dashboard, see [Using the CMDB Integrations Dashboard](#).

Data Mapping

Data from the New Relic data sources is mapped and transformed into the ServiceNow CMDB Configuration Item (CI) class definitions using the Robust Transform Engine (RTE). Data is inserted into the ServiceNow CMDB using the Identification and Reconciliation Engine (IRE).

You can use the IntegrationHub ETL app to view the data maps. See [IntegrationHub ETL \(3.2\)](#) for more information.

The following data sources are included for New Relic:

- SG-New Relic Application Services
- SG-New Relic Applications
- SG-New Relic Disks
- SG-New Relic Hosts
- SG_New Relic Networks

When you complete the guided setup, you can configure the integration to periodically pull data from the New Relic application. The data is loaded into the following staging tables:

- SG-New Relic Applications
[sn_newrelic_integ_sg_new_relic_applications]
- SG-New Relic Application Services
[sn_newrelic_integ_application_services]
- SG-New Relic Disks [sn_newrelic_integ_disks]
- SG-New Relic Hosts [sn_newrelic_integ_hosts]

- SG_New Relic Networks [sn_newrelic_integ_networks]

The data is then inserted into the following target tables:

- Application [cmdb_ci_appl]
- Application service [cmdb_ci_service_calculated]
- Disk [cmdb_ci_disk]
- IP Address[cmdb_ci_ip_address]
- Network Adapter [cmdb_ci_network_adapter]
- Running Process [cmdb_running_process]
- Server [cmdb_ci_server]
- Software [cmdb_ci_spkg]
- Software Installation [cmdb_sam_sw_install]
- Software Instance [cmdb_software_instance]

For more information on where data is saved when pulling data from New Relic, see [CMDB classes targeted](#).

Set up scheduled import jobs to pull in data from New Relic into your CMDB.

Before you begin

Dependencies and requirements:

- The [Integration Commons for CMDB](#) store app, which is automatically installed.
- The [CMDB CI Class Models store app](#) store app, which is automatically installed.
- The ITOM Discovery License plugin (com.snc.itom.discovery.license). You must activate this plugin.
- ITOM Licensing plugin (com.snc.itom.license). For more information, see [Request Discovery](#).

- The Datastream Action plugin (`com.glide.hub.action_type.datastream`), which is automatically installed.
- Observability Commons for CMDB (`sn_observability`), which is only required for event ingestion. This must be installed prior to installing the connector for Event Management to work. For more information, see [Observability Commons for CMDB](#) on the ServiceNow Store.

Note: If you have an earlier version of the Service Graph Connector for Observability - New Relic, then don't migrate data from the old connector. You must uninstall the previous version and run the new integration.

Starting with the San Diego release, embedded help content won't be visible on the default Polaris theme in the Next Experience. You must activate the embedded help content by completing the following steps:

1. Select a task section in the guided setup, and then click **Configure**.



2. In the menu bar, click the help icon ().

Role required: admin

Procedure

1. Ensure that the application scope is set to the Service Graph Connector for Observability - New Relic application by using the application picker.
For more information, see [Application picker](#).
2. Navigate to **All > Service Graph Connectors > New Relic > Setup**.
3. On the Getting started page, select **Get Started**.
4. Configure the authentication credentials to send requests to the New Relic application.
 - a. Configure your New Relic connection and credentials.
 - a. In the Configure the Connection section of the New Relic Integration with CMDB page, select **Get Started**.

- b. For the Configure connection and credentials task, select **Configure**.
- c. On the Connections page of the Flow Designer, select **Configure** for the **NewRelicConnectionAlias** connection that is available by default.
- d. On the form, review and modify the fields.

Configure Connection form

Field	Description
Connection Information	
Connection Name	Name to uniquely identify the connection record. For example, NewRelicConnectionAlias .
Connection URL	Base URL to connect to the New Relic application in the following format: https://api.newrelic.com/graphql Note: This field is automatically set to the URL to connect to the NerdGraph API in the New Relic application. Leave the field value as is.
Credential Information	
API Key	NerdGraph API token used for authentication on the New Relic application.
Account ID	Account ID associated with the New Relic credential.

e. Select **Configure Connection**.

Note: The Service Graph Connector for Observability

- New Relic supports connection to a single New Relic instance only. So, you can use the single default connection only.

f. Return to the Configure the connection task page using the back button for your browser.

g. Set the Configure connection and credentials task to complete by clicking **Mark as Complete**.

b. Test the NerdGraph API connection to import data from the New Relic application.

a. In the Configure the Connection section of the New Relic Integration with CMDB page, select **Continue**.

b. For the Test Connection task, select **Configure**.

c. Click the **Test Load 20 Records** related link.

The Test Connection dialog box opens displaying the import progress.

d. When the progress state changes to **Complete**, click **X** to close the Test Connection dialog box and return to the setup.

e. Set the Test Connection task to complete by clicking **Mark as Complete**.

5. Configure the webhooks for Observability New Relic and turn on alerts for unmatched configuration items (CIs).

a. Configure the webhooks for Observability New Relic.

a. In the Configure Observability section of the New Relic Integration with CMDB page, select **Get started**.

b. For the Configure the webhooks task, select **Configure**.

c. In the SG-New Relic Webhooks list, click **New** to add a New Relic webhook.

d. On the form, fill in the fields.

New record form

Field	Description
Name	Name of the New Relic webhook.
Connection Alias	Search for and select the connection and credential alias you created in step 4.

- e. Click **Submit**.
- f. Set the Configure the webhooks task to complete by clicking **Mark as Complete**.
- b. Enable alerts for configuration items (CIs) that aren't available in the CMDB.
 - a. In the Configure Observability section of the New Relic Integration with CMDB page, select **Continue**.
 - b. For the Turn on alerts for unmatched CI task, select **Configure**.
 - c. On the System Property form, fill in the fields to create the sn_newrelic_integ.alerts_for_unmatched_ci.enabled system property and set its value to true.
- For more information, see [Add a system property](#).
- d. Click **Submit**.
- e. Set the Turn on alerts for unmatched CI task to complete by clicking **Mark as Complete**.
6. Configure the scheduled jobs to import data from the New Relic application.
 - a. In the Set up scheduled data imports section of the New Relic Integration with CMDB page, select **Get started**.
 - b. For the Configure Scheduled Data Imports task, select **Configure**.
 - c. Select the scheduled job that you want to activate.

- d. On the Scheduled Data Import form, verify the field values for the scheduled job.
For more information, see [Schedule a data import](#).
- e. Click **Execute Now**.
- f. Repeat the steps [6.c](#) to [6.e](#) for each scheduled job for data import.
- g. Click **X** to close the Configure Scheduled Data Imports window and return to the setup page.
- h. Set the Configure Scheduled Data Imports task to complete by clicking **Mark as Complete** in the guided setup.

When you complete the guided setup, you can configure the integration to periodically pull data from New Relic. The data is saved in tables that extend from the Configuration item [cmdb_ci] table.

The following attributes in the Server [cmdb_ci_server] table are populated by collected data:

Attribute label	Attribute name
CPU core count	cpu_core_count
Disk space (GB)	disk_space
DNS Domain	dns_domain
Fully qualified domain name	fqdn
Host name	host_name
Operating System	os

Relationships created for Server

Parent class	Relationship type	Child class
Server [cmdb_ci_server]	Contains::Contained by	Disk [cmdb_ci_disk]

Parent class	Relationship type	Child class
Server [cmdb_ci_server]	Owns::Owned by	IP Address [cmdb_ci_ip_address]
Server [cmdb_ci_server]	Owns::Owned by	Network Adapter [cmdb_ci_network_adapter]

The following attributes in the Calculated Application Service [cmdb_ci_service_calculated] table are populated by collected data:

Attribute label	Attribute name
Name	name
Metadata	metadata
Service Populator Status	populator_status
Service Type	type

The following attributes in the IP Address [cmdb_ci_ip_address] table are populated by collected data:

Attribute label	Attribute name
IP Address	ip_address
IP version	ip_version
Name	name
Nic	nic

Relationship created for IP Address

Parent class	Relationship type	Child class
IP Address [cmdb_ci_ip_address]	Reference	Network Adapter [cmdb_ci_network_adapter]

The following attributes in the Software Instance [cmdb_software_instance] table are populated by collected data:

Attribute label	Attribute name
Installed on	installed_on
Name	name

Relationship created for Software Instance

Parent class	Relationship type	Child class
Software Instance [cmdb_software_instance]	Reference	Server [cmdb_ci_server]

The following attributes in the Software [cmdb_ci_spkg] table are populated by collected data:

Attribute label	Attribute name
Key	key
Nic	nic
Name	name

Relationship created for Software

Parent class	Relationship type	Child class
Software [cmdb_ci_spkg]	Reference	Software Instance [cmdb_software_instance]

The following attributes in the Disk [cmdb_ci_disk] table are populated by collected data:

Attribute label	Attribute name
Device ID	device_id

Attribute label	Attribute name
Free disk space (GB)	free_space
Computer	computer
Name	name
Disk space (GB)	disk_space
File system	file_system

Relationship created for Disk

Parent class	Relationship type	Child class
Disk [cmdb_ci_disk]	Hosted on::Hosts	Server [cmdb_ci_server]

The following attributes in the Network Adapter [cmdb_ci_network_adapter] table are populated by collected data:

Attribute label	Attribute name
Configuration Item	cmdb_ci
MAC Address	mac_address
Name	name

Relationship created for Network Adapter

Parent class	Relationship type	Child class
Network Adapter [cmdb_ci_network_adapter]	Reference	Server [cmdb_ci_server]

The following attributes in the Application [cmdb_ci_appl] table are populated by collected data:

Attribute label	Attribute name
Class	sys_class_name
Name	name
Running process command	running_process_command

Relationship created for Application

Parent class	Relationship type	Child class
Application [cmdb_ci_appl]	Runs on::Runs	Server [cmdb_ci_server]

Service Graph Connector for ExtraHop (2.0.3)

Use the Service Graph Connector for ExtraHop to pull data from the ExtraHop application into your ServiceNow instance.

Important: Starting with the Vancouver release, the ServiceNow hosted Service Graph Connector for Extrahop is being prepared for future deprecation. It will be hidden and no longer activated on new instances but will continue to be supported. The ExtraHop hosted Service Graph Connector for Extrahop provides the latest experience for this functionality. For details, see the [Deprecation Process \[KB0867184\]](#) article in the Now Support Knowledge Base.

The Service Graph Connector for ExtraHop provides real-time network visibility across your enterprise by implementing stream processing, so that you can transform your network data into structured wire data.

The Service Graph Connector for ExtraHop pulls network visibility data into the ServiceNow® Configuration Management Database (CMDB) application. The connector enriches discovered device data and establishes relationships between devices based on network traffic flow.

Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

System requirements and supported versions

Dependencies and requirements:

- MID Server that is installed on Linux or Windows, unless the ExtraHop appliance is publicly accessible.
- ExtraHop Discover appliance with firmware version 7.2 or later with a user account that has unlimited privileges.
- Supported versions: ExtraHop v7.9.
- Supported ServiceNow versions:
 - Quebec
 - Rome
 - San Diego
 - Tokyo

Use cases

The following are examples on how you can use the Service Graph Connector:

- Identification of network interactions between CIs.
- Discovery of network traffic flow data between computers or other types of hardware.
- Creation of relationships between devices. The relationships are based on network traffic flow.

Guided setup

The guided setup for the Service Graph Connector for ExtraHop provides an organized sequence of tasks to configure the integration on your instance. To access the guided setup, see [Configure guided setup](#).

CMDB Integrations Dashboard

The Integration Commons for CMDB store app provides a dashboard with a central view of the status, processing results, and processing errors of all installed Service Graph Connectors. You can see metrics for all integration runs. You can also filter the view to a specific CMDB integration, a specific time duration, or a specific integration run. For more details about monitoring ExtraHop integrations in the CMDB Integrations Dashboard, see [Using the CMDB Integrations Dashboard](#).

Data mappings

Data from data sources in the ExtraHop application is mapped and transformed into the ServiceNow CMDB Configuration Item (CI) class definitions using the Robust Transform Engine (RTE). Data is inserted into the ServiceNow CMDB using the Identification and Reconciliation Engine (IRE).

When you complete the guided setup, you can configure the integration to periodically pull data from the ExtraHop application. The data is loaded into the following staging tables:

- ExtraHop Computer [sn_extrahop_integr_computer]
- ExtraHop Network Activity [sn_extrahop_integr_activity]

The data is then inserted into the following target tables:

- CI Relationship [cmdb_rel_ci]
- Hardware [cmdb_ci_hardware]
- IP Address [cmdb_ci_ip_address]
- Network Adapter [cmdb_ci_network_adapter]

Set up a REST message and scheduled jobs to import ExtraHop data into your CMDB.

Before you begin

Important: Starting with the Vancouver release, the ServiceNow hosted Service Graph Connector for ExtraHop is being prepared for future deprecation. It will be hidden and no longer activated on new instances but will continue to be supported. The ExtraHop hosted Service Graph Connector for ExtraHop provides the latest experience for this functionality. For details, see the [Deprecation Process \[KB0867184\]](#) article in the Now Support Knowledge Base.

To use this Service Graph Connector, you need a subscription to a Subscription Unit that is based in the ServiceNow® IT Operations Management (ITOM) Visibility application or in the ITOM Discovery application. As defined in the section titled "Managed IT Resource Types" in [ServiceNow Subscription Unit Overview](#), for managed IT resources that are created or modified in the CMDB by this Service Graph Connector, but that are not yet managed by [ITOM Visibility or ITOM Discovery](#), these resources will increase Subscription Unit consumption from that application. Review your current Subscription Unit consumption within ITOM Visibility or ITOM Discovery to ensure available capacity.

Dependencies and requirements:

- The [Integration Commons for CMDB](#) store app, which is automatically installed.
- The [CMDB CI Class Models](#) store app, which is automatically installed.
- ITOM Licensing plugin (com.snc.itom.license). For more information, see [Request Discovery](#).
- ITOM Discovery License plugin (com.snc.itom.discovery.license). You must activate this plugin.
- MID Server that is installed on Linux or Windows, unless the ExtraHop appliance is publicly accessible.
- ExtraHop Discover appliance with firmware version 7.2 or later, with a user account that has unlimited privileges.
- To connect to the ExtraHop application, configure an API key. For more information, see [ExtraHop REST API Guide](#), specifically, see the "ExtraHop API requirements" section.

Starting with the San Diego release, embedded help content will not be visible on the default Polaris theme in the Next Experience. You must activate the embedded help content by completing the following steps:

1. Select a task section in the guided setup, and then click **Configure**.
2. In the menu bar, click the help icon (?

Roles required: admin

Procedure

1. Navigate to **All > Service Graph Connector ExtraHop > Setup**.
2. On the Getting started page, click **Get Started**.
3. On the Service Graph Connector for the ExtraHop page, in the Configure REST message section, select the task **Configure ExtraHop REST message**.
4. Configure a REST message to use when sending requests to the ExtraHop API.
 - a. On the next page, in the Configure ExtraHop REST message task section, click **Configure**.
 - b. On the form, fill in the fields.

REST Message form

Field	Description
Name	Descriptive name for this REST Message. This field is automatically set.
Endpoint	URL to the web service API endpoint. Set the field to a base URL. For example, https://myextrahop.com .

Field	Description
	<p>Note: Updating the Endpoint base URL also updates the base URLs for all HTTP methods that are associated with the ExtraHop REST message. In the HTTP Request tab, update the Authorization header to use your API key in the Value field.</p>
Description	Description for this REST Message. This field is automatically set.
Application	Application that contains this message. The field is automatically set.
Authentication type	Type of authentication to apply to HTTP requests. The field is automatically set.
Use mutual authentication	Option to use multiple authentication by authenticating HTTP requests. Mutual authentication cannot be used with a MID Server.

- c. Click **Update** if necessary.
 - d. In the Configure ExtraHop REST message task section, click **Mark as Complete**.
5. Test the connection to the ExtraHop Computer API.
- a. In the Test Computer connection task section, click **Configure**.
 - b. On the form, fill in the fields.

HTTP Method form

Field	Description
Name	Unique identifier for this HTTP method. This field is automatically set.
Endpoint	URL to the web service API endpoint
Use MID Server	MID Server that sends this HTTP request. Using a MID Server is not compatible with mutual authentication.
REST Message	REST message record that this method is based on. This field is automatically set.
HTTP method	HTTP method that is implemented by this method. This field is automatically set.
Application	Application that contains this method. This field is automatically set.
Authentication type	Type of authentication to apply to HTTP requests.
Use mutual authentication	Option to use multiple authentication by authenticating HTTP requests. Mutual authentication cannot be used with a MID Server.

- c. Select the **HTTP Request** tab and fill out the **Use MID Server** field.
- d. Click the **Authentication** tab and then click the **Test** related link.
Testing the connection takes a few moments. When the test is complete, the page is refreshed and shows the test results.

e. Select **Mark as Complete**.

Note: The connection is successful if the **HTTP Status** field is set to **200**. If there are any errors in the **Error Message** field, then the connection failed and further troubleshooting is required.

f. Click **Update** if necessary.g. In the Test Computer connection task section, click **Mark as Complete**.

6. Test the connection to the ExtraHop Network Activity Create API.

a. In the Test Network Activity Create connection task section, click **Configure**.

b. On the form, fill in the fields.

HTTP Method form

Field	Description
Name	Unique identifier for this HTTP method.
Endpoint	URL to the web service API endpoint
Use MID Server	MID Server that sends this HTTP request. Using a MID Server is not compatible with mutual authentication.
REST Message	REST message that this method is based on. This field is automatically set.
HTTP method	HTTP method that is implemented by this method. This field is automatically set.

Field	Description
Application	Application that contains this record. This field is automatically set.
Authentication type	Type of authentication to apply to HTTP requests. This field is automatically set.
Use mutual authentication	Option to use multiple authentication by authenticating HTTP requests. Mutual authentication cannot be used with a MID Server.

- c. Click the **Authentication** tab and then click the **Test** related link. Testing the connection takes a few moments. When the test is complete, the page is refreshed and shows the test results.
 - d. Select **Mark as Complete**.

Note: The connection is successful if the **HTTP Status** field is set to **200**. If there are any errors in the **Error Message** field, then the connection failed and further troubleshooting is required.
 - e. Click **Update** if necessary.
 - f. In the Test Network Activity Create connection task section, click **Mark as Complete**.
7. Set up the scheduled import jobs.
- a. On the Service Graph Connector for ExtraHop page, in the Set up scheduled import jobs section, select the task **Configure Computer scheduled job**.
 - b. In the Configure Computer scheduled job task section, click **Configure**.
 - c. On the form, fill in the fields.

Scheduled Data Import form

Field	Description
Name	Name of the scheduled job.
Data source	Data source record that defines the data to import.
Run as	Option to run the scheduled job with the credentials of the specified user.
Active	Option to activate the scheduled job. Select this option.
Concurrent Import	Function that loads the data from multiple import sets. The function then processes and transforms the data concurrently.
Partition Method	Partition method for the concurrent import set.
Partition Size	Import set size for early scheduling.
Execute pre-import script	Option to specify a script to run before the import is performed.
Execute post-import script	Option to specify a script to run after the import is performed.
Application	Application that contains this scheduled job.
Run	Frequency of running the import.

Field	Description
Conditional	Conditions under which this job is executed.

- d. Click **Update** if necessary.
 - e. In the Configure Computer scheduled job task section, click **Mark as Complete**.
8. Configure the Network Activity scheduled job.
- a. On the Service Graph Connector for ExtraHop page, in the Set up scheduled import jobs section, select the task **Configure Network Activity scheduled job**.
 - b. In the Configure Network Activity scheduled job task section, click **Configure**.
 - c. On the form, fill in the fields.

Scheduled Data Import form

Field	Description
Name	Name of the scheduled job.
Data source	Data source record that defines the data to import.
Run as	Option to run the scheduled job with the credentials of the specified user.
Active	Option to activate the scheduled job. Select this option.
Concurrent Import	Function that loads the data from multiple import sets. The function then processes and transforms the data concurrently.

Field	Description
Partition Method	Partition method for the concurrent import set.
Partition Size	Import set size for early scheduling.
Execute pre-import script	Option to specify a script to run before the import is performed.
Execute post-import script	Option to specify a script to run after the import is performed.
Application	Application that contains this scheduled job.
Run	Frequency of running the import.
Conditional	Conditions under which this job is executed.

- d. Click **Update** if necessary then **Mark as Complete**.

When you complete the guided setup, you can configure the integration to periodically pull data from ExtraHop. The data is saved in tables that extend from the Configuration item [cmdb_ci] table.

Important: Starting with the Vancouver release, the ServiceNow hosted Service Graph Connector for ExtraHop is being prepared for future deprecation. It will be hidden and no longer activated on new instances but will continue to be supported. The ExtraHop hosted Service Graph Connector for ExtraHop provides the latest experience for this functionality. For details, see the [Deprecation Process \[KB0867184\]](#) article in the Now Support Knowledge Base.

The following attributes in the Hardware [cmdb_ci_hardware] table are populated by collected data:

Attribute label	Attribute name
Name	name
Manufacturer	manufacturer

Relationships created for Hardware

Parent class	Relationship type	Child class
Hardware [cmdb_ci_hardware]	Owns::Owned by	IP Address [cmdb_ci_ip_address]
Hardware [cmdb_ci_hardware]	Owns::Owned by	Network Adapter [cmdb_ci_network_adapter]
Hardware [cmdb_ci_hardware]	Receives data from::Sends data to	Hardware [cmdb_ci_hardware]

The following attributes in the IP Address [cmdb_ci_ip_address] table are populated by collected data:

Attribute label	Attribute name
IP Address	ip_address
IP version	ip_version
Name	name
Nic	nic

Relationship created for IP Address

Parent class	Relationship type	Child class
IP Address [cmdb_ci_ip_address]	Reference	Network Adapter [cmdb_ci_network_adapter]

The following attributes in the Network Adapter [cmdb_ci_network_adapter] table are populated by collected data:

Attribute label	Attribute name
Name	name
Configuration Item	cmdb_ci
MAC Address	mac_address

Relationship created for Network Adapter

Parent class	Relationship type	Child class
Network Adapter [cmdb_ci_network_adapter]	Reference	Hardware [cmdb_ci_hardware]

Service Graph Connector for GCP (1.3.1)

Use the Service Graph Connector for GCP to ingest CMDB data from a Google Cloud Platform (GCP) installation using REST APIs.

Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

Supported versions

- GCP version: Last tested on July 03, 2023
- Supported ServiceNow versions:
 - San Diego
 - Tokyo
 - Utah

- Vancouver

Use cases

You can use the Service Graph Connector for GCP to get visibility into cloud resource identities, relationships, and state in real-time.

Guided setup

The guided setup for the Service Graph Connector for GCP provides an organized sequence of tasks to configure the integration on your instance. To access the guided setup, see [Configure guided setup](#).

CMDB integrations dashboard

The Integration Commons for CMDB store app provides a dashboard with a central view of the status, processing results, and processing errors of all installed integrations. You can see metrics for all integration runs. You can filter the view to a specific CMDB integration, a specific time duration, or a specific integration run. For more details about monitoring GCP integrations in the CMDB Integrations Dashboard, see [Using the CMDB Integrations Dashboard](#).

Data mapping

Data from the GCP data sources is mapped and transformed into the ServiceNow CMDB Configuration Item (CI) class definitions using the Robust Transform Engine (RTE). Data is inserted into the ServiceNow CMDB using the Identification and Reconciliation Engine (IRE).

When you complete the guided setup, you can configure the integration to periodically pull data from the GCP application.

You can use the IntegrationHub ETL app to view the data maps. See [IntegrationHub ETL \(3.2\)](#) for more information.

The following tables lists the data sources included for GCP and the corresponding staging tables where the imported data is loaded.

Data sources and staging tables for GCP

Data source	Staging table
SG-GCP Cloud Database	SG-GCP Cloud Database [sn_gcp_integ_sg_gcp_cloud_database]
SG-GCP Cloud Function	SG-GCP Cloud Function [sn_gcp_integ_sg_gcp_cloud_function]
SG-GCP Cloud Object Storage	SG-GCP Cloud Object Storage [sn_gcp_integ_sg_gcp_cloud_object_storage]
SG-GCP Folder	SG-GCP Folder [sn_gcp_integ_sg_gcp_folder]
SG-GCP Hardware Type	SG-GCP Hardware Type [sn_gcp_integ_sg_gcp_hardware_type]
SG-GCP Image	SG-GCP Image [sn_gcp_integ_sg_gcp_image]
SG-GCP Kubernetes Cluster	SG-GCP Kubernetes Cluster [sn_gcp_integ_sg_gcp_kubernetes_cluster]
SG-GCP Kubernetes Deployment	SG-GCP Kubernetes Deployment [sn_gcp_integ_sg_gcp_kubernetes_deployment]
SG-GCP Kubernetes Namespace	SG-GCP Kubernetes Namespace [sn_gcp_integ_sg_gcp_kubernetes_namespace]
SG-GCP Kubernetes Node	SG-GCP Kubernetes Node [sn_gcp_integ_sg_gcp_kubernetes_node]

Data source	Staging table
SG-GCP Kubernetes Pod	SG-GCP Kubernetes Pod [sn_gcp_integ_sg_gcp_kubernetes_pod]
SG-GCP Kubernetes Replicaset	SG-GCP Kubernetes Replicaset [sn_gcp_integ_sg_gcp_kubernetes_replicaset]
SG-GCP Kubernetes Service	SG-GCP Kubernetes Service [sn_gcp_integ_sg_gcp_kubernetes_service]
SG-GCP Load Balancer	SG-GCP Load Balancer [sn_gcp_integ_sg_gcp_load_balancer]
SG-GCP Load Balancer Health Service	SG-GCP Load Balancer Health Service [sn_gcp_integ_sg_gcp_load_balancer_health_service]
SG-GCP Load Balancer Pool	SG-GCP Load Balancer Pool [sn_gcp_integ_sg_gcp_load_balancer_pool]
SG-GCP Load Balancer Pool Member	SG-GCP Load Balancer Pool Member [sn_gcp_integ_sg_gcp_load_balancer_pool_member]
SG-GCP Load Balancer Service	SG-GCP Load Balancer Service [sn_gcp_integ_sg_gcp_load_balancer_service]
SG-GCP Network	SG-GCP Network [sn_gcp_integ_sg_gcp_network]
SG-GCP Organization	SG-GCP Organization [sn_gcp_integ_sg_gcp_organization]

Data source	Staging table
SG-GCP Project	SG-GCP Project [sn_gcp_integ_sg_gcp_project]
SG-GCP Security Group	SG-GCP Security Group [sn_gcp_integ_sg_gcp_security_group]
SG-GCP Software Inventory	SG-GCP Software Inventory [sn_gcp_integ_sg_gcp_software_inventory]
SG-GCP Storage Volume	SG-GCP Storage Volume [sn_gcp_integ_sg_gcp_storage_volume]
SG-GCP Storage Volume Snapshot	SG-GCP Storage Volume Snapshot [sn_gcp_integ_sg_gcp_storage_vol_snapshot]
SG-GCP Subnet	SG-GCP Subnet [sn_gcp_integ_sg_gcp_subnet]
SG-GCP VM Hw Consolidation	SG-GCP VM Hw Consolidation [sn_gcp_integ_sg_gcp_vm_hw_consolidation]
SG-GCP VM Instance	SG-GCP VM Instance [sn_gcp_integ_sg_gcp_vm_instance]

The imported data from the staging tables is then inserted into the following target tables:

- Availability Zone [cmdb_ci_availability_zone]
- Cloud DataBase [cmdb_ci_cloud_database]
- Cloud Disk Type [cmdb_ci_disk_type]
- Cloud Function [cmdb_ci_cloud_function]

- Cloud Load Balancer [cmdb_ci_cloud_load_balancer]
- Cloud Load Balancer Health Service [cmdb_ci_lb_health_service]
- Cloud Mgmt Network Interface [cmdb_ci_nic]
- Cloud Network [cmdb_ci_network]
- Cloud Object Storage [cmdb_ci_cloud_object_storage]
- Cloud Organizations [cmdb_ci_cloud_org]
- Cloud Service Account [cmdb_ci_cloud_service_account]
- Cloud Subnet [cmdb_ci_cloud_subnet]
- Compute Security Group [cmdb_ci_compute_security_group]
- Docker Container [cmdb_ci_docker_container]
- Docker Image [cmdb_ci_docker_image]
- Google Datacenter [cmdb_ci_google_datacenter]
- Google Organization Folder [cmdb_ci_gcp_folder]
- Google Organization Project [cmdb_ci_gcp_project]
- Hardware Type [cmdb_ci_compute_template]
- Image [cmdb_ci_os_template]
- IP Address [cmdb_ci_ip_address]
- Kubernetes Volume [cmdb_ci_kubernetes_volume]
- Kubernetes Pod [cmdb_ci_kubernetes_pod]
- Kubernetes ReplicaSet [cmdb_ci_kubernetes_replicaset]
- Kubernetes Service [cmdb_ci_kubernetes_service]
- Kubernetes Namespace [cmdb_ci_kubernetes_namespace]
- Kubernetes Cluster [cmdb_ci_kubernetes_cluster]
- Kubernetes Node [cmdb_ci_kubernetes_node]

- Kubernetes Deployment [cmdb_ci_kubernetes_deployment]
- Load Balancer Pool [cmdb_ci_lb_pool]
- Load Balancer Pool Member [cmdb_ci_lb_pool_member]
- Load Balancer Service [cmdb_ci_lb_service]
- Network ACL [cmdb_ci_network_acl]
- Network ACL Rule [cmdb_ci_network_acl_rule]
- Server [cmdb_ci_server]
- Software Installation [cmdb_sam_sw_install] (If SAM is installed.)
- Software Instance [cmdb_software_instance] (If SAM is not installed.)
- Software [cmdb_ci_spkg] (If SAM is not installed.)
- Storage Mapping [cmdb_ci_storage_mapping]
- Storage Volume [cmdb_ci_storage_volume]
- Storage Volume Snapshot [cmdb_ci_storage_vol_snapshot]
- Virtual Machine Instance [cmdb_ci_vm_instance]
- Cloud Disk Type [cmdb_ci_disk_type]
- Docker Container [cmdb_ci_docker_container]
- Docker Image [cmdb_ci_docker_image]
- Block Endpoint [cmdb_ci_endpoint_block]
- VNIC Endpoint [cmdb_ci_endpoint_vnic]

For more information on where data is saved when pulling data from GCP, see [CMDB classes targeted](#).

Set up scheduled import jobs to pull in data from Google Cloud Platform (GCP) into your CMDB.

Before you begin

Dependencies and requirements:

- The [Integration Commons for CMDB](#) store app, which is automatically installed.
- The [CMDB CI Class Models store app](#) store app, which is automatically installed.
- The ITOM Discovery License plugin (com.snc.item.discovery.license). You must activate this plugin.
- ITOM Licensing plugin (com.snc.item.license). For more information, see [Request Discovery](#).
- The Datastream Action plugin (com.glide.hub.action_type.datastream), which is automatically installed.
- Observability Commons for CMDB (sn_observability), which is only required for event ingestion. This app must be installed prior to installing the connector for Event Management to work. For more information, see [Observability Commons for CMDB](#) on the ServiceNow Store.

Note: If you have an earlier version of the Service Graph Connector for GCP, then don't migrate data from the old connector. You must uninstall the previous version and run the new integration.

Starting with the San Diego release, embedded help content won't be visible on the default Polaris theme in the Next Experience. You must activate the embedded help content by completing the following steps:

1. Select a task section in the guided setup, and then click **Configure**.
2. In the menu bar, click the help icon (?

Role required: admin

Procedure

1. Ensure that the application scope is set to the Service Graph Connector for GCP application by using the application picker. For more information, see [Application picker](#).

2. Navigate to **All > Service Graph Connectors > GCP > Setup**.
3. On the Getting started page, select **Get Started**.
4. Configure the connection to send requests to the GCP application.
 - a. In the Configure the Connection and Credentials section of the Service Graph Connector for GCP page, select **Get Started**.
 - b. Set up the GCP environment and create a Java KeyStore (JKS) certificate to encrypt the security certificates obtained from a GCP application.

Make a note of the destination keystore password. You need to specify this password while importing the JKS certificate into the Service Graph Connector for GCP application.

For instructions, see the [Service Graph Connector for GCP - Setup Instructions \[KB1220598\]](#) article in the Now Support Knowledge Base.

Note: After you have set up the GCP environment, return to the guided setup and set the GCP Setup Instructions task in the Configure the Connection and Credentials section to complete by clicking **Mark as Complete**.

- c. Create an X.509 certificate to associate the JKS certificate for the GCP application with the Service Graph Connector for GCP.
 - a. For the Create X.509 certificate task, select **Configure**.
 - b. On the form that opens in a new tab, fill in the fields.

X.509 Certificate form

Field	Description
Name	Name of the X.509 certificate. For example, SG-GCP-509Certificate-Org1.
Notify on expiration	Users to be notified when the certificate expires. If no users are selected, the logged-in user is added by

Field	Description
	default, along with the last two logged-in users with the administrator role.
Warn in days to expire	Number of days to send a notification before the certificate expires.
Active	Option to activate the certificate.
Type	Certificate container that is automatically set to Java Key Store . Leave the field value set to Java Key Store .
Expires in days	Number of days until the certificate expires.
Key store password	Password to access the JKS certificate as noted down in the previous step .
Short description	Description of the X.509 certificate.

- c. Click the manage attachments icon ().
- d. Click **Choose file** to browse and upload the keystore.p12 file for the JKS certificate you created in step 4.b.
- e. Close the dialog box.
- f. Click the **Validate Stores/Certificates** related link.
- g. On successful validation, click **Update** to return to the guided setup page.
- h. Set the Create X.509 certificate task to complete by clicking **Mark as Complete**.

- d. Create a JSON Web Token (JWT) signing key from the X.509 certificate credentials.
 - a. For the Create JWT Key task, select **Configure**.
 - b. On the form that opens in a new tab, fill in the fields.

JWT Keys Certificate form

Field	Description
Name	Name of the JWT signing key. For example, SG-GCP-Keys-Org1.
Signing Keystore	Name of the X.509 certificate that you created in step 4.C.
Key Id	ID to identify which JWT signing key is used when multiple keys are used to sign tokens.
Application	Name of the application using the JWT signing key. This field is automatically set to Service Graph Connector for GCP .
Signing Algorithm	Algorithm to sign with the JWT signing key that is automatically set to RSA 256 . Leave the field value set to RSA 256 .
Signing Key	Password associated with the JWT signing key.
Active	Option to activate the JWT signing key.

- c. Click **Submit** to return to the guided setup page.

- d. Set the Create JWT Key task to complete by clicking **Mark as Complete**.
- e. Add a JWT provider for the GCP application.
 - a. For the Create JWT Provider task, select **Configure**.
 - b. On the form that opens in a new tab, fill in the fields.

JWT Provider form

Field	Description
Name	Name to uniquely identify the JWT provider.
Expiry Interval (sec)	Number of seconds that indicate the lifespan of the JWT provider token.
Signing Configuration	Name of the JWT signing key you created in step 4.d.

- c. Click **Submit** to return to the guided setup page.
- d. Set the Create JWT Provider task to complete by clicking **Mark as Complete**.
- f. Create a mapping between the JWT credential and GCP organization or project from which the data is imported.
 - a. For the Create Organization-Credential Mapping task, select **Configure**.
 - b. On the form that opens in a new tab, fill in the fields.

SG-GCP Organization Credential Setup form

Field	Description
JWT Provider	Name of the JWT provider you created in step 4.e.

Field	Description
Organization Id	ID of the organization associated with the GCP application.
Service Account	ServiceNow service account associated with the GCP application.
Discovery Scope	<p>Discovery scope of the GCP application. The available options are:</p> <p>Organization Select Organization when the ServiceNow service account has access to data within the GCP organization.</p> <p>Projects Select Projects when the ServiceNow service account has access to data within GCP projects only.</p>

- c. Click **Submit** to return to the guided setup page.
- d. Set the Create Organization-Credential Mapping task to complete by clicking **Mark as Complete**.
- g. Map the organization credential with the pre-defined data sources.
 - a. For the Create Organization-Credential Mapping task, select **Configure**.
 - b. In the **Organization-Credential Record** field of the form that opens in a new tab, click the lookup using list icon () to

select the name of the JWT provider you mapped with the GCP organization in step [4.f.](#)

- c. Click **Execute mapping** to process all the data sources and scheduled jobs to import data from the GCP application.
 - d. Return to the guided setup page and set the Create Organization-Credential Mapping task to complete by clicking **Mark as Complete**.
 - h. Configure the scheduled jobs to import data from the GCP application.
 - a. For the Configure the Scheduled Imports task, select **Configure**.
 - b. Select the scheduled job that you want to activate.
 - c. On the Scheduled Data Import form, verify the field values for the scheduled job.
- For more information, see [Schedule a data import](#).
- d. Click **Execute Now**.
 - e. Repeat the steps [4.h.ii](#) to [4.h.iv](#) for each scheduled job for data import.
 - f. Click the back icon (<) to return to the guided setup page.
 - g. Set the Configure the Scheduled Imports task to complete by clicking **Mark as Complete** in the guided setup

5. (Optional) Add multiple GCP instances.

- a. In the Add Multiple Instances section of the Service Graph Connector for GCP page, select **Get Started**.
- b. Create data sources for the new GCP connection.
 - a. Ensure that you have edit permissions for the Datasource [sys_data_source] table.
 - b. For the Update Data Source Access task, click **Configure**.

- c. To edit the record, select the **Global** application scope from the application picker.
 - d. In the Application Access related list of the Data Source form that opens in a new tab, select the **Can create**, **Can update**, and **Can delete** check boxes.
 - e. Click **Update**.
 - f. Click the back icon (<) to return to the guided setup page tab.
 - g. From the application picker, select the **Service Graph Connector for GCP** application scope.
 - h. Set the Update Data Source Access task to complete by clicking **Mark as Complete**.
- c. Create a scheduled import job for the new GCP connection.
 - a. Ensure that you have edit permissions for the Scheduled data import [scheduled_import_set] table.
 - b. For the Update Scheduled Data Import Access task, click **Configure**.
 - c. To edit the record, select the **Global** application scope from the application picker.
 - d. In the Application Access related list of the Scheduled Data Import form that opens in a new tab, select the **Can create**, **Can update**, and **Can delete** check boxes.
 - e. Click **Update**.
 - f. Click the back icon (<) to return to the guided setup page tab.
 - g. From the application picker, select the **Service Graph Connector for GCP** application scope.
 - h. Set the Update Scheduled Data Import Access task to complete by clicking **Mark as Complete**.
 - d. Clear the cache on the Data Source [sys_data_source] and Scheduled Data Imports [scheduled_import_set] tables.

- a. For the Clear Cache for Data Source and Scheduled Data Imports tables task, click **Configure**.
- b. To edit the record, select the **Global** application scope from the application picker.
- c. In the **Run script** field, enter the following code:

```
GlideTableManager.invalidateTable("sys_data_source");
GlideCacheManager.flushTable("sys_data_source");
GlideTableManager.invalidateTable("scheduled_import_set");
GlideCacheManager.flushTable("scheduled_import_set");
GlideTableManager.invalidateTable("sys_db_object");
GlideCacheManager.flushTable("sys_db_object");
```

- d. Click **Run script**.
- e. Click the back icon (<) to return to the guided setup page tab.
- f. From the application picker, select the **Service Graph Connector for GCP** application scope.
- g. Set the Clear Cache for Data Source and Scheduled Data Imports tables task to complete by clicking **Mark as Complete**.
- e. To create an X.509 certificate for the new GCP instance, repeat the step [4.c](#), and then mark the Create X.509 certificate task to complete by clicking **Mark as Complete**.
- f. To create a JWT key for the new GCP instance, repeat the step [4.d](#), and then mark the Create JWT Key task to complete by clicking **Mark as Complete**.
- g. To create a JWT provider for the new GCP instance, repeat the step [4.e](#), and then mark the Create JWT Provider task to complete by clicking **Mark as Complete**.
- h. To create an organization-credential mapping for the new GCP instance, repeat the step [4.f](#), and then mark the Create

Organization-Credential Mapping task to complete by clicking **Mark as Complete**.

- i. Generate data sources and scheduled imports for the organization credential mapping associated with the new GCP connection.
 - a. For the Generate Data Sources and Scheduled Imports task, select **Configure**.
 - b. In the **Data source and Scheduled Import name prefix** field of the form that opens in a new tab, enter a prefix for the data sources and scheduled job for the new GCP connection.
 - c. In the **Organization-Credential Record** field of the form that opens in a new tab, click the lookup using list icon () to select the name of the JWT provider you mapped with the GCP organization.
 - d. Click **Generate Data source and Scheduled Import** to generate and process all the data sources and scheduled jobs for the new connection.
 - e. Return to the guided setup page and set the Generate Data Sources and Scheduled Imports task to complete by clicking **Mark as Complete**.
- j. To configure the scheduled jobs to import data from the new GCP instance, repeat the step **4.h**, and then mark the Configure the Scheduled Imports task to complete by clicking **Mark as Complete**.

When you complete the guided setup, you can configure the integration to periodically pull data from GCP. The data is saved in tables that extend from the Configuration item [cmdb_ci] table.

Availability Zone [cmdb_ci_availability_zone]

The following attributes in the Availability Zone [cmdb_ci_availability_zone] table are populated by collected data:

Attribute label	Attribute name
Name	name

Attribute label	Attribute name
Object ID	object_id

Block Endpoint [cmdb_ci_endpoint_block]

The following attributes in the Block Endpoint [cmdb_ci_endpoint_block] table are populated by collected data:

Attribute label	Attribute name
Host	host
Name	name

Cloud DataBase [cmdb_ci_cloud_database]

The following attributes in the Cloud DataBase [cmdb_ci_cloud_database] table are populated by collected data:

Attribute label	Attribute name
Fully qualified domain name	fqdn
Install status	install_status
IP Address	ip_address
Name	name
Object ID	object_id
Type	type
Version	version

Relationship created for Cloud DataBase

Parent class	Relationship type	Child class
Cloud DataBase [cmdb_ci_cloud_data base]	Hosted on::Hosts	Google Datacenter [cmdb_ci_google_dat acenter]
Cloud DataBase [cmdb_ci_cloud_data base]	Reference	Key Value [cmdb_key_value]

Cloud Disk Type [cmdb_ci_disk_type]

The following attributes in the Cloud Disk Type [cmdb_ci_disk_type] table are populated by collected data:

Attribute label	Attribute name
Name	name
Object ID	object_id

Relationships created for Cloud Disk Type

Parent class	Relationship type	Child class
Cloud Disk Type [cmdb_ci_disk_type]	Hosted on::Hosts	Google Datacenter [cmdb_ci_google_dat acenter]
Cloud Disk Type [cmdb_ci_disk_type]	Reference	Key Value [cmdb_key_value]

Cloud Function [cmdb_ci_cloud_function]

The following attributes in the Cloud Function [cmdb_ci_cloud_function] table are populated by collected data:

Attribute label	Attribute name
CodeSha256	codesha256
Function Last Modified	function_last_modified
Language	language
Name	name
Object ID	object_id
Version	version

Relationship created for Cloud Function

Parent class	Relationship type	Child class
Cloud Function [cmdb_ci_cloud_function]	Hosted on::Hosts	Google Datacenter [cmdb_ci_google_datacenter]
Cloud Function [cmdb_ci_cloud_function]	Reference	Key Value [cmdb_key_value]

Cloud Load Balancer [cmdb_ci_cloud_load_balancer]

The following attributes in the Cloud Load Balancer [cmdb_ci_cloud_load_balancer] table are populated by collected data:

Attribute label	Attribute name
Name	name
Object ID	object_id

Relationship created for Cloud Load Balancer

Parent class	Relationship type	Child class
Cloud Load Balancer [cmdb_ci_cloud_load_balancer]	Hosted on::Hosts	Google Datacenter [cmdb_ci_google_datacenter]
Cloud Load Balancer [cmdb_ci_cloud_load_balancer]	Reference	Key Value [cmdb_key_value]

Cloud Load Balancer Health Service [cmdb_ci_lb_health_service]

The following attributes in the Cloud Load Balancer Health Service [cmdb_ci_lb_health_service] table are populated by collected data:

Attribute label	Attribute name
Healthy threshold	healthy_threshold
Interval in seconds	check_interval_sec
Monitor type protocol	monitor_type
Name	name
Object ID	object_id
Port	port
Request path	request_path
Timeout in seconds	timeout_sec
Unhealthy threshold	unhealthy_threshold

Relationship created for Cloud Load Balancer Health Service

Parent class	Relationship type	Child class
Cloud Load Balancer Health Service [cmdb_ci_lb_health_service]	Hosted on::Hosts	Cloud Service Account [cmdb_ci_cloud_service_account]
Cloud Load Balancer Health Service [cmdb_ci_lb_health_service]	Contains::Contained by	Cloud Load Balancer [cmdb_ci_cloud_load_balancer]

Cloud Mgmt Network Interface [cmdb_ci_nic]

The following attributes in the Cloud Mgmt Network Interface [cmdb_ci_nic] table are populated by collected data:

Attribute label	Attribute name
IP Address	ip_address
Name	name
Object ID	object_id

Relationship created for Cloud Mgmt Network Interface

Parent class	Relationship type	Child class
Cloud Mgmt Network Interface [cmdb_ci_nic]	Owns::Owned by	IP Address [cmdb_ci_ip_address]

Cloud Network [cmdb_ci_network]

The following attributes in the Cloud Network [cmdb_ci_network] table are populated by collected data:

Attribute label	Attribute name
Name	name
Object ID	object_id

Relationships created for Cloud Network

Parent class	Relationship type	Child class
Cloud Network [cmdb_ci_network]	Hosted on::Hosts	Cloud Service Account [cmdb_ci_cloud_service_account]
Cloud Network [cmdb_ci_network]	Contains::Contained by	Cloud Subnet [cmdb_ci_cloud_subnet]
Cloud Network [cmdb_ci_network]	Contains::Contained by	Network ACL [cmdb_ci_network_acl]

Cloud Object Storage [cmdb_ci_cloud_object_storage]

The following attributes in the Cloud Object Storage [cmdb_ci_cloud_object_storage] table are populated by collected data:

Attribute label	Attribute name
Cloud Provider	cloud_provider
Name	name
Object ID	object_id
Service Name	service_name

Relationship created for Cloud Object Storage

Parent class	Relationship type	Child class
Cloud Object Storage [cmdb_ci_cloud_object_storage]	Hosted on::Hosts	Google Datacenter [cmdb_ci_google_datacenter]
Cloud Object Storage [cmdb_ci_cloud_object_storage]	Reference	Key Value [cmdb_key_value]

Cloud Organizations [cmdb_ci_cloud_org]

The following attributes in the Cloud Organizations [cmdb_ci_cloud_org] table are populated by collected data:

Attribute label	Attribute name
Install status	install_status
Name	name
Object ID	object_id
Operational status	operational_status
Time	time

Relationship created for Cloud Organizations

Parent class	Relationship type	Child class
Cloud Organizations [cmdb_ci_cloud_org]	Contains::Contained by	Google Organization Folder [cmdb_ci_gcp_folder]

Cloud Service Account [cmdb_ci_cloud_service_account]

The following attributes in the Cloud Service Account [cmdb_ci_cloud_service_account] table are populated by collected data:

Attribute label	Attribute name
Account Id	account_id
Datacenter Type	datacenter_type
Install status	install_status
Name	name
Object ID	object_id
Operational status	operational_status

Cloud Subnet [cmdb_ci_cloud_subnet]

The following attributes in the Cloud Subnet [cmdb_ci_cloud_subnet] table are populated by collected data:

Attribute label	Attribute name
Name	name
CIDR	cidr
Object ID	object_id

Relationships created for Cloud Subnet

Parent class	Relationship type	Child class
Cloud Subnet [cmdb_ci_cloud_subnet]	Contains::Contained by	Kubernetes Cluster [cmdb_ci_kubernetes_cluster]
Cloud Subnet [cmdb_ci_cloud_subnet]	Reference	Key Value [cmdb_key_value]

Compute Security Group [cmdb_ci_compute_security_group]

The following attributes in the Compute Security Group [cmdb_ci_compute_security_group] table are populated by collected data:

Attribute label	Attribute name
Name	name
Object ID	object_id

Relationships created for Compute Security Group

Parent class	Relationship type	Child class
Compute Security Group [cmdb_ci_compute_security_group]	Hosted on::Hosts	Cloud Service Account [cmdb_ci_cloud_service_account]
Compute Security Group [cmdb_ci_compute_security_group]	Reference	Key Value [cmdb_key_value]

Docker Container [cmdb_ci_docker_container]

The following attributes in the Docker Container [cmdb_ci_docker_container] table are populated by collected data:

Attribute label	Attribute name
Container id	container_id
Command	command
Container created	container_created_at
Image id	image_id
Name	name

Attribute label	Attribute name
Status	status

Docker Image [cmdb_ci_docker_image]

The following attributes in the Docker Image [cmdb_ci_docker_image] table are populated by collected data:

Attribute label	Attribute name
Image id	image_id
Name	name

Google Datacenter [cmdb_ci_google_datacenter]

The following attributes in the Google Datacenter [cmdb_ci_google_datacenter] table are populated by collected data:

Attribute label	Attribute name
Install status	install_status
Name	name
Object ID	object_id
Operational status	operational_status
Region	region

Relationships created for Google Datacenter

Parent class	Relationship type	Child class
Google Datacenter [cmdb_ci_google_datacenter]	Contains::Contained by	Availability Zone [cmdb_ci_availability_zone]

Parent class	Relationship type	Child class
Google Datacenter [cmdb_ci_google_dat acenter]	Hosted on::Hosts	Cloud Service Account [cmdb_ci_cloud_servi ce_account]

Google Organization Folder [cmdb_ci_gcp_folder]

The following attributes in the Google Organization Folder [cmdb_ci_gcp_folder] table are populated by collected data:

Attribute label	Attribute name
Install status	install_status
Name	name
Object ID	object_id
Operational status	operational_status
Parent Id	parent_id
Parent Type	parent_type

Google Organization Project [cmdb_ci_gcp_project]

The following attributes in the Google Organization Project [cmdb_ci_gcp_project] table are populated by collected data:

Attribute label	Attribute name
Install status	install_status
Name	name
Object ID	object_id
Operational status	operational_status

Attribute label	Attribute name
Parent	parent_ci
Parent Id	parent_id
Parent Type	parent_type
Project Id	project_id
Time	time

Relationships created for Google Organization Project

Parent class	Relationship type	Child class
Google Organization Project [cmdb_ci_gcp_project]	Reference	Google Organization Folder [cmdb_ci_gcp_folder]
Google Organization Project [cmdb_ci_gcp_project]	Reference	Cloud Organizations [cmdb_ci_cloud_org]

Hardware Type [cmdb_ci_compute_template]

The following attributes in the Hardware Type [cmdb_ci_compute_template] table are populated by collected data:

Attribute label	Attribute name
Local Storage GB	local_storage_gb
Memory MB	memory_mb
Name	name
Object ID	object_id
vCPUs	vcpus

Relationship created for Hardware Type

Parent class	Relationship type	Child class
Hardware Type [cmdb_ci_compute_template]	Hosted on::Hosts	Google Datacenter [cmdb_ci_google_datacenter]

Image [cmdb_ci_os_template]

The following attributes in the Image [cmdb_ci_os_template] table are populated by collected data:

Attribute label	Attribute name
Install status	install_status
Name	name
Object ID	object_id
Operational status	operational_status

Relationships created for Image

Parent class	Relationship type	Child class
Image [cmdb_ci_os_template]	Hosted on::Hosts	Cloud Service Account [cmdb_ci_cloud_service_account]
Image [cmdb_ci_os_template]	Reference	Key Value [cmdb_key_value]

IP Address [cmdb_ci_ip_address]

The following attributes in the IP Address [cmdb_ci_ip_address] table are populated by collected data:

Attribute label	Attribute name
IP Address	ip_address
IP version	ip_version
Name	name

Key Value [cmdb_key_value]

The following attributes in the Key Value [cmdb_key_value] table are populated by collected data:

Attribute label	Attribute name
Key	key
Value	value

Kubernetes Cluster [cmdb_ci_kubernetes_cluster]

The following attributes in the Kubernetes Cluster [cmdb_ci_kubernetes_cluster] table are populated by collected data:

Attribute label	Attribute name
Kubernetes UID	k8s_uid
Name	name

Relationships created for Kubernetes Cluster

Parent class	Relationship type	Child class
Kubernetes Cluster [cmdb_ci_kubernetes_cluster]	Hosted on::Hosts	Google Datacenter [cmdb_ci_google_datacenter]
Kubernetes Cluster [cmdb_ci_kubernetes_cluster]	Cluster of::Cluster	Kubernetes Node [cmdb_ci_kubernetes_node]

Parent class	Relationship type	Child class
Kubernetes Cluster [cmdb_ci_kubernetes_cluster]	Contains::Contained by	Kubernetes Pod [cmdb_ci_kubernetes_pod]
Kubernetes Cluster [cmdb_ci_kubernetes_cluster]	Contains::Contained by	Kubernetes Namespace [cmdb_ci_kubernetes_namespace]
Kubernetes Cluster [cmdb_ci_kubernetes_cluster]	Contains::Contained by	Kubernetes Service [cmdb_ci_kubernetes_service]
Kubernetes Cluster [cmdb_ci_kubernetes_cluster]	Reference	Key Value [cmdb_key_value]

Kubernetes Deployment [cmdb_ci_kubernetes_deployment]

The following attributes in the Kubernetes Deployment [cmdb_ci_kubernetes_deployment] table are populated by collected data:

Attribute label	Attribute name
Available Replicas	available_replicas
Desired Replicas	desired_replicas
Kubernetes Cluster	cluster
Kubernetes UID	k8s_uid
Name	name
Namespace	namespace
Total Replicas	total_replicas
Unavailable Replicas	unavailable_replicas

Attribute label	Attribute name
Updated Replicas	updated_replicas

Relationships created for Kubernetes Deployment

Parent class	Relationship type	Child class
Kubernetes Deployment [cmdb_ci_kubernetes_deployment]	Hosted on::Hosts	Kubernetes Cluster [cmdb_ci_kubernetes_cluster]
Kubernetes Deployment [cmdb_ci_kubernetes_deployment]	Reference	Kubernetes Cluster [cmdb_ci_kubernetes_cluster]
Kubernetes Deployment [cmdb_ci_kubernetes_deployment]	Reference	Key Value [cmdb_key_value]

Kubernetes Namespace [cmdb_ci_kubernetes_namespace]

The following attributes in the Kubernetes Namespace [cmdb_ci_kubernetes_namespace] table are populated by collected data:

Attribute label	Attribute name
Kubernetes UID	k8s_uid
Name	name
Namespace	namespace

Relationship created for Kubernetes Namespace

Parent class	Relationship type	Child class
Kubernetes Namespace [cmdb_ci_kubernetes_namespace]	Reference	Key Value [cmdb_key_value]

Kubernetes Node [cmdb_ci_kubernetes_node]

The following attributes in the Kubernetes Node [cmdb_ci_kubernetes_node] table are populated by collected data:

Attribute label	Attribute name
IP Address	ip_address
Kubernetes Cluster	cluster
Kubernetes UID	k8s_uid
Name	name
Namespace	namespace

Relationships created for Kubernetes Node

Parent class	Relationship type	Child class
Kubernetes Node [cmdb_ci_kubernetes_node]	Hosted on::Hosts	Google Datacenter [cmdb_ci_google_datacenter]
Kubernetes Node [cmdb_ci_kubernetes_node]	Reference	Kubernetes Cluster [cmdb_ci_kubernetes_cluster]

Kubernetes Pod [cmdb_ci_kubernetes_pod]

The following attributes in the Kubernetes Pod [cmdb_ci_kubernetes_pod] table are populated by collected data:

Attribute label	Attribute name
IP Address	ip_address
Kubernetes UID	k8s_uid
Name	name
Namespace	namespace
Resource version	resource_version

Relationships created for Kubernetes Pod

Parent class	Relationship type	Child class
Kubernetes Pod [cmdb_ci_kubernetes_pod]	Contains::Contained by	Docker Image [cmdb_ci_docker_image]
Kubernetes Pod [cmdb_ci_kubernetes_pod]	Contains::Contained by	Kubernetes Volume [cmdb_ci_kubernetes_volume]
Kubernetes Pod [cmdb_ci_kubernetes_pod]	Contains::Contained by	Docker Container [cmdb_ci_docker_container]
Kubernetes Pod [cmdb_ci_kubernetes_pod]	Reference	Key Value [cmdb_key_value]

Kubernetes ReplicaSet [cmdb_ci_kubernetes_replicaset]

The following attributes in the Kubernetes ReplicaSet [cmdb_ci_kubernetes_replicaset] table are populated by collected data:

Attribute label	Attribute name
Desired Replicas	desired_replicas

Attribute label	Attribute name
Kubernetes Cluster	cluster
Kubernetes UID	k8s_uid
Name	name
Namespace	namespace
SelfLink	self_link
Total Replicas	total_replicas

Relationships created for Kubernetes ReplicaSet

Parent class	Relationship type	Child class
Kubernetes ReplicaSet [cmdb_ci_kubernetes_replicaset]	Hosted on::Hosts	Kubernetes Cluster [cmdb_ci_kubernetes_cluster]
Kubernetes ReplicaSet [cmdb_ci_kubernetes_replicaset]	Reference	Kubernetes Cluster [cmdb_ci_kubernetes_cluster]
Kubernetes ReplicaSet [cmdb_ci_kubernetes_replicaset]	Reference	Key Value [cmdb_key_value]

Kubernetes Service [cmdb_ci_kubernetes_service]

The following attributes in the Kubernetes Service [cmdb_ci_kubernetes_service] table are populated by collected data:

Attribute label	Attribute name
IP Address	ip_address
Kubernetes UID	k8s_uid
Name	name

Attribute label	Attribute name
Namespace	namespace

Relationship created for Kubernetes Service

Parent class	Relationship type	Child class
Kubernetes Service [cmdb_ci_kubernetes_service]	Reference	Key Value [cmdb_key_value]

Kubernetes Volume [cmdb_ci_kubernetes_volume]

The following attributes in the Kubernetes Volume [cmdb_ci_kubernetes_volume] table are populated by collected data:

Attribute label	Attribute name
Kubernetes UID	k8s_uid
Mount Path	mount_path
Name	name
Namespace	namespace
Volume ID	volume_id

Load Balancer Pool [cmdb_ci_lb_pool]

The following attributes in the Load Balancer Pool [cmdb_ci_lb_pool] table are populated by collected data:

Attribute label	Attribute name
Name	name
Object ID	object_id

Relationships created for Load Balancer Pool

Parent class	Relationship type	Child class
Load Balancer Pool [cmdb_ci_lb_pool]	Hosted on::Hosts	Google Datacenter [cmdb_ci_google_datacenter]
Load Balancer Pool [cmdb_ci_lb_pool]	Owns::Owned by	Load Balancer Pool Member [cmdb_ci_lb_pool_member]
Load Balancer Pool [cmdb_ci_lb_pool]	Reference	Key Value [cmdb_key_value]

Load Balancer Pool Member [cmdb_ci_lb_pool_member]

The following attributes in the Load Balancer Pool Member [cmdb_ci_lb_pool_member] table are populated by collected data:

Attribute label	Attribute name
Install status	install_status
Name	name
Object ID	object_id
Operational status	operational_status

Load Balancer Service [cmdb_ci_lb_service]

The following attributes in the Load Balancer Service [cmdb_ci_lb_service] table are populated by collected data:

Attribute label	Attribute name
Listener Protocol	listener_protocol
Name	name

Attribute label	Attribute name
Object ID	object_id
Port	port

Relationships created for Load Balancer Service

Parent class	Relationship type	Child class
Load Balancer Service [cmdb_ci_lb_service]	Contains::Contained by	Load Balancer Pool [cmdb_ci_lb_pool]
Load Balancer Service [cmdb_ci_lb_service]	Hosted on::Hosts	Cloud Load Balancer [cmdb_ci_cloud_load_balancer]

Network ACL [cmdb_ci_network_acl]

The following attributes in the Network ACL [cmdb_ci_network_acl] table are populated by collected data:

Attribute label	Attribute name
Name	name
Object ID	object_id

Relationships created for Network ACL

Parent class	Relationship type	Child class
Network ACL [cmdb_ci_network_acl]	Contains::Contained by	Network ACL Rule [cmdb_ci_network_acl_rule]
Network ACL [cmdb_ci_network_acl]	Reference	Key Value [cmdb_key_value]

Network ACL Rule [cmdb_ci_network_acl_rule]

The following attributes in the Network ACL Rule [cmdb_ci_network_acl_rule] table are populated by collected data:

Attribute label	Attribute name
Allow Deny	allow_deny
Allowed\Denied Traffic	allowed_denied_traffic
Destination Ranges	destination_ranges
Name	name
Outbound	is_outbound
Source Ranges	source_ranges
Target Tags	target_tags

Relationship created for Network ACL Rule

Parent class	Relationship type	Child class
Network ACL Rule [cmdb_ci_network_acl_rule]	Reference	Key Value [cmdb_key_value]

Server [cmdb_ci_server]

The following attribute in the Server [cmdb_ci_server] table is populated by collected data:

Attribute label	Attribute name
Name	name
Operating System	os
OS Version	os_version

Relationships created for Server

Parent class	Relationship type	Child class
Server [cmdb_ci_server]	Virtualized by::Virtualizes	Virtual Machine Instance [cmdb_ci_vm_instance]
Server [cmdb_ci_server]	Owns::Owned by	IP Address [cmdb_ci_ip_address]
Server [cmdb_ci_server]	Reference	Software Installation [cmdb_sam_sw_install]
Server [cmdb_ci_server]	Reference	Key Value [cmdb_key_value]

Software [cmdb_ci_spkg]

The following attributes in the Software [cmdb_ci_spkg] table are populated by collected data when the Software Asset Management (SAM) application isn't installed:

Attribute label	Attribute name
Discovery source	discovery_source
Key	key
Manufacturer	manufacturer
Name	name
Version	version

Relationship created for Software

Parent class	Relationship type	Child class
Software [cmdb_ci_spkg]	Reference	Software Instance [cmdb_software_instance]

Software Installation [cmdb_sam_sw_install]

The following attributes in the Software Installation [cmdb_sam_sw_install] table are populated by collected data when the SAM application is installed:

Attribute label	Attribute name
Discovery source	discovery_source
Display name	display_name
Installed on	installed_on
Last scanned	last_scanned
Publisher	publisher
Version	version

Relationship created for Software Installation

Parent class	Relationship type	Child class
Software Installation [cmdb_sam_sw_install]	Reference	Server [cmdb_ci_server]

Software Instance [cmdb_software_instance]

The following attributes in the Software Instance [cmdb_software_instance] table are populated by collected data when the SAM application isn't installed:

Attribute label	Attribute name
Install date	install_date
Installed on	installed_on
Name	name

Relationship created for Software Instance

Parent class	Relationship type	Child class
Software Instance [cmdb_software_instance]	Reference	Server [cmdb_ci_server]

Storage Mapping [cmdb_ci_storage_mapping]

The following attributes in the Storage Mapping [cmdb_ci_storage_mapping] table are populated by collected data:

Attribute label	Attribute name
Mapping Type	mapping_type
Name	name
Object ID	object_id

Storage Volume [cmdb_ci_storage_volume]

The following attributes in the Storage Volume [cmdb_ci_storage_volume] table are populated by collected data:

Attribute label	Attribute name
Install status	install_status
Name	name

Attribute label	Attribute name
Object ID	object_id
Operational status	operational_status
Size bytes	size_bytes
State	state
Storage type	storage_type
Volume ID	volume_id

Relationships created for Storage Volume

Parent class	Relationship type	Child class
Storage Volume [cmdb_ci_storage_volume]	Hosted on::Hosts	Google Datacenter [cmdb_ci_google_datacenter]
Storage Volume [cmdb_ci_storage_volume]	Reference	Key Value [cmdb_key_value]

Storage Volume Snapshot [cmdb_ci_storage_vol_snapshot]

The following attributes in the Storage Volume Snapshot [cmdb_ci_storage_vol_snapshot] table are populated by collected data:

Attribute label	Attribute name
Install status	install_status
Name	name
Object ID	object_id
Operational status	operational_status
Parent ID	parent_id

Attribute label	Attribute name
Size (GB)	size
State	state
Volume Name	volume_name

Relationships created for Storage Volume Snapshot

Parent class	Relationship type	Child class
Storage Volume Snapshot [cmdb_ci_storage_vol_snapshot]	Hosted on::Hosts	Google Datacenter [cmdb_ci_google_datacenter]
Storage Volume Snapshot [cmdb_ci_storage_vol_snapshot]	Reference	Key Value [cmdb_key_value]

Virtual Machine Instance [cmdb_ci_vm_instance]

The following attributes in the Virtual Machine Instance [cmdb_ci_vm_instance] table are populated by collected data:

Attribute label	Attribute name
CPUs	cpus
Disks	disks
Disks size (GB)	disks_size
Install status	install_status
Memory (MB)	memory
Name	name
Network adapters	nics

Attribute label	Attribute name
Object ID	object_id
Operational status	operational_status
State	state
VM Instance ID	vm_inst_id

Relationships created for Virtual Machine Instance

Parent class	Relationship type	Child class
Virtual Machine Instance [cmdb_ci_vm_instance]	Provisioned From::Provisioned	Hardware Type [cmdb_ci_compute_template]
Virtual Machine Instance [cmdb_ci_vm_instance]	Contains::Contained by	Cloud Subnet [cmdb_ci_cloud_subnet]
Virtual Machine Instance [cmdb_ci_vm_instance]	Hosted on::Hosts	Google Datacenter [cmdb_ci_google_datacenter]
Virtual Machine Instance [cmdb_ci_vm_instance]	Contains::Contained by	Cloud Mgmt Network Interface [cmdb_ci_nic]
Virtual Machine Instance [cmdb_ci_vm_instance]	Contains::Contained by	Storage Mapping [cmdb_ci_storage_mapping]
Virtual Machine Instance	Provisioned From::Provisioned	Image [cmdb_ci_os_template]

Parent class	Relationship type	Child class
[cmdb_ci_vm_instance]		
Virtual Machine Instance [cmdb_ci_vm_instance]	Use End Point To::Use End Point From	Block Endpoint [cmdb_ci_endpoint_block]
Virtual Machine Instance [cmdb_ci_vm_instance]	Hosted on::Hosts	Google Organization Project [cmdb_ci_gcp_project]
Virtual Machine Instance [cmdb_ci_vm_instance]	Reference	Key Value [cmdb_key_value]

VNIC Endpoint [cmdb_ci_endpoint_vnic]

The following attributes in the VNIC Endpoint [cmdb_ci_endpoint_vnic] table are populated by collected data:

Attribute label	Attribute name
Host	host
Name	name
Object ID	object_id

Relationship created for VNIC Endpoint

Parent class	Relationship type	Child class
VNIC Endpoint [cmdb_ci_endpoint_vnic]	Implement End Point To::Implement End Point From	Cloud Mgmt Network Interface [cmdb_ci_nic]

Service Graph Connector for Infoblox (1.1.0)

Use the Service Graph Connector for Infoblox to pull data from an Infoblox instance into your ServiceNow instance.

Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

Supported versions

- Supported versions: Infoblox API v2.11.2
- Supported ServiceNow versions:
 - San Diego
 - Tokyo
 - Utah
 - Vancouver

Use cases

The following are examples on how you can use the Service Graph Connector for Infoblox for different ServiceNow® applications:

- Configure the Infoblox connection for connecting to an Infoblox instance.
- Create IP address management (IPAM) tasks when subnets are added or deleted.
- Schedule periodic synchronization of IPAM CIs by configuring scheduled data imports jobs.
- View imported IPAM CIs and IPAM tasks.

Guided setup

The guided setup for the Service Graph Connector for Infoblox provides an organized sequence of tasks to configure the integration on your instance. To access the guided setup, see [Configure guided setup](#).

Note: Users with the import_scheduler and import_admin roles can modify the scripts in scheduled imports and potentially escalate their privilege to admin. So, be careful when granting the import_scheduler and import_admin roles to any users.

CMDB integrations dashboard

The Integration Commons for CMDB store app provides a dashboard with a central view of the status, processing results, and processing errors of all installed integrations. You can see metrics for all integration runs. You can filter the view to a specific CMDB integration, a specific time duration, or a specific integration run. For more details about monitoring Infoblox integrations in the CMDB Integrations Dashboard, see [Using the CMDB Integrations Dashboard](#).

Data mapping

Data from the Infoblox data source is mapped and transformed into the ServiceNow CMDB configuration item (CI) class definitions using the Robust Transform Engine (RTE). Data is inserted into the ServiceNow CMDB using the Identification and Reconciliation Engine (IRE).

When you complete the guided setup, you can configure the integration to pull data periodically from Infoblox.

You can use the IntegrationHub ETL app to view the data maps. See [IntegrationHub ETL \(3.2\)](#) for more information.

The following data sources are included for the Infoblox app:

SG-Infoblox IP Pool

Imports all the IP pools, networks, and subnets from the Infoblox instance, loads the imported data in the SG-Infoblox IP Address [sn_infoblox_integ_sg_infoblox_ip_address] staging table, and then populates the IP Pool [cmdb_ci_ip_pool] target table.

SG-Infoblox IP Address

Imports all the IPv4 and IPv6 data from Infoblox instance, loads the imported data in the SG Infoblox Connection [sn_infoblox_integ_sg_infoblox_connection] staging table, and then populates the Allocated IP Address [cmdb_ci_allocated_ip_address], IP Network Subnet [cmdb_ci_ip_network_subnet], and Managed Network [cmdb_ci_managed_network] target tables.

SG-Infoblox DNS Alias

Imports all the DNS aliases from the Infoblox instance, loads the imported data in the SG-Infoblox DNS Alias [sn_infoblox_integ_sg_infoblox_dns_alias] staging table, and then populates the DNS Alias [cmdb_ci_dns_alias] target table.

For more information on where data is saved when pulling data from Infoblox, see [CMDB classes targeted](#).

Pull in data from Infoblox into your CMDB.

Before you begin

Dependencies and requirements:

- The [Integration Commons for CMDB](#) store app, which is automatically installed.
- The [CMDB CI Class Models](#) store app store app, which is automatically installed.
- The ITOM Discovery License plugin (com.snc.item.discovery.license). You must activate this plugin.
- ITOM Licensing plugin (com.snc.item.license). For more information, see [Request Discovery](#).
- The Datastream Action plugin (com.glide.hub.action_type.datastream), which is automatically installed.
- Observability Commons for CMDB (sn_observability), which is only required for event ingestion. This must be installed prior to installing the connector for Event Management to work. For more information, see [Observability Commons for CMDB](#) on the ServiceNow Store.

Note: If you have an earlier version of the Service Graph Connector for Infoblox, then don't migrate data from the old connector. You must uninstall the previous version and run the new integration.

Starting with the San Diego release, embedded help content won't be visible on the default Polaris theme in the Next Experience. You must activate the embedded help content by completing the following steps:

1. Select a task section in the guided setup, and then click **Configure**.
2. In the menu bar, click the help icon (?

Role required: admin

Procedure

1. Ensure that the application scope is set to the Service Graph Connector for Infoblox application by using the application picker. For more information, see [Application picker](#).
2. Navigate to **All > Service Graph Connectors > Infoblox > Setup**.
3. On the Getting started page, select **Get Started**.
4. Configure the authentication credentials and HTTP connection details for sending requests to the Infoblox API.
 - a. Configure your Infoblox authentication credentials.
 - a. In the Configure the connection section of the Service Graph Connector for Infoblox page, select **Get Started**.
 - b. For the Configure Infoblox authentication credentials task, select **Configure** to open the Basic Auth Credentials page opens in a new browser tab..
 - c. In the **Name** field, enter a name for the authentication. For example, **Infoblox Credential**.
 - d. In the **User name** field, enter the user name that is used to authenticate the HTTP request when this authentication profile is enabled.

- e. In the **Password** field, enter the password for the user name that is used to authenticate the HTTP request.
 - f. Click **Update** to return to the guided setup page.
 - g. Set the Configure Infoblox authentication credentials task to complete by clicking **Mark as Complete**.
- b. Configure the Infoblox connection settings.
 - a. For the Configure Infoblox HTTP connection task, select **Configure** to open the HTTP(s) Connection page in a new browser tab.
 - b. Review the fields and in the **Name** field and enter the Infoblox instance name.
 - c. Use the URL builder to build the connection string or in the **Connection URL** field, enter the Infoblox base URL in the following format: `https://<base-URL>`.
 - d. In the **api_version** field of the Attributes section, enter the [version of WAPI](#) you're using.
 - e. In the **network_view** field of the Attributes section, enter the network views for which data is to be imported.

For multiple views, separate the views with commas. For example: `view1,view2`. If you leave this field empty, the data sources import data from all the network views.

- f. Click **Update** to return to the guided setup page.
- g. Set the Configure Infoblox HTTP connection task to complete by clicking **Mark as Complete**.

For more information about the Infoblox API, see the [Infoblox Developer documentation](#).

- c. Configure the connection properties.
 - a. For the Configure connection properties task, select **Configure** to open the Service Graph Connections page in a new browser tab.

- b. To review and modify the property details, click a property from the **Property** column in the Service Graph Connection Properties related list.
- c. Click **Update** on the Service Graph Connection Properties page.
- d. Click **Update** on the Service Graph Connections page to return to the guided setup page.
- e. Set the Configure connection properties task to complete by clicking **Mark as Complete**.
- d. Test the Infoblox API connection to import data from the Infoblox application.
 - a. In the Configure the connection section of the Service Graph Connector for Infoblox page, select **Continue**.
 - b. For the Test the connection task, select **Configure** to open the Service Graph Connections page in a new browser tab.
 - c. Click **Test Connection**.
 - d. When the **Status** field is set to **Success**, select **Update** to close the Test the connection dialog box and return to the guided setup page.

If any of the tests have errors, follow the suggestions for remediation.

- e. Set the Test the connection task to complete by clicking **Mark as Complete**.
5. Configure the IP address management tasks and user groups.
 - a. Enable properties to create an IP address management task when deleting or inserting a network.
 - a. In the Configure IP address management tasks section of the Service Graph Connector for Infoblox page, select **Get Started**.
 - b. For the Enable creating a task task, select **Configure** to open the Enable create Task page in a new browser tab.

- c. Select the check box for the **Create a task when a network is deleted** and **Create a task when a network is inserted** fields to enable properties for creating IP address management tasks when deleting and inserting a network, respectively.
 - d. Click **Save**.
 - e. Close the Enable create Task page tab and return to the guided setup page.
 - f. Set the Enable creating a task task to complete by clicking **Mark as Complete**.
- b. Configure the user group for IP address management tasks.
 - a. For the Configure the user group for IP address management tasks task, select **Configure** to open the SG Infoblox Connection page in a new browser tab.
 - b. In the **Connection alias** field, review and change the connection alias.
 - c. In the **Task user group** field, change the assignment group for the connection alias.

Note: By default, the user group for the IP address management tasks is assigned to the IP Address Management user group. To change the default assignment group, change the task user group for the connection alias.
 - d. Click **Update** to return to the guided setup page.
 - e. Set the Configure the user group for IP address management tasks task to complete by clicking **Mark as Complete**.
6. Configure the scheduled jobs to import data from the Infoblox application.
 - a. In the Set up scheduled import jobs section of the Service Graph Connector for Infoblox page, select **Get started**.
 - b. For the Configure the scheduled jobs task, select **Configure** to open the Scheduled Data Imports page in a new browser tab.

- c. From the **Name** column, select the scheduled job that you want to activate.
- d. On the Scheduled Data Import form, verify the field values for the scheduled job.
For more information, see [Schedule a data import](#).
- e. Click **Execute Now**.
- f. Repeat the steps [6.c](#) to [6.e](#) for each scheduled job for data import.
- g. Close the Scheduled Data Imports page tab and return to the guided setup page.
- h. Set the Configure the scheduled jobs task to complete by clicking **Mark as Complete** in the guided setup.

When you complete the guided setup, you can configure the integration to pull data periodically from Infoblox. The data is saved in tables that extend from the Configuration item [cmdb_ci] table.

Allocated IP Address [cmdb_ci_allocated_ip_address]

The following attributes in the Allocated IP Address [cmdb_ci_allocated_ip_address] table are populated by collected data:

Attribute label	Attribute name
IP Address	ip_address
Managed Network	managed_network
Is Broadcast	is_broadcast
Is Conflict	is_conflict
Is DHCP	is_dhcp
Is DNS	is_dns
Is Managed	is_managed

Attribute label	Attribute name
Is Reserved	is_reserved
Name	name

Relationship created for Allocated IP Address

Parent class	Relationship type	Child class
Allocated IP Address [cmdb_ci_allocated_ip_address]	Reference	Managed Network [cmdb_ci_managed_network]

IP Network Subnet [cmdb_ci_ip_network_subnet]

The following attributes in the IP Network Subnet [cmdb_ci_ip_network_subnet] table are populated by collected data:

Attribute label	Attribute name
Name	name
Parent Pool	parent_pool
CIDR	cidr

Relationship created for IP Network Subnet

Parent class	Relationship type	Child class
IP Network Subnet [cmdb_ci_ip_network_subnet]	Members::Member of subnet	Allocated IP Address [cmdb_ci_allocated_ip_address]

IP Pool [cmdb_ci_ip_pool]

The following attributes in the IP Pool [cmdb_ci_ip_pool] table are populated by collected data:

Attribute label	Attribute name
Name	name
Parent Pool	parent_pool
CIDR	cidr

Managed Network [cmdb_ci_managed_network]

The following attribute in the Managed Network [cmdb_ci_managed_network] table is populated by collected data:

Attribute label	Attribute name
Name	name

DNS Alias [cmdb_ci_dns_alias]

The following attributes in the DNS Alias [cmdb_ci_dns_alias] table is populated by collected data:

Attribute label	Attribute name
Name	name

Service Graph Connector for Microsoft Intune (2.3.0)

Use the Service Graph Connector for Microsoft Intune to pull data from the Microsoft Intune application into your ServiceNow instance.

The Service Graph Connector for Microsoft Intune pulls data from mobile devices, computers and software applications into the ServiceNow® Configuration Management Database (CMDB) application. The integration provides greater visibility into mobile devices, computers and related software applications that run on them.

Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

Supported versions

- Supported versions:
 - Microsoft Intune Graph API v1.0
 - Microsoft Intune Graph API Beta
- Supported ServiceNow versions:
 - San Diego
 - Tokyo
 - Utah
 - Vancouver

Use Cases

The following are examples on how you can use the Service Graph Connector for different ServiceNow® applications:

- [IT Operations Management \(ITOM\) Visibility](#)
 - Detailed hardware and application inventory for Android, Apple, and Windows mobile devices. The inventory can be used with or without Software Asset Management (SAM).
 - Compliance tracking for mobile devices. You can build your own device (BYOD) or use corporate-owned devices.
- [IT Service Management \(ITSM\)](#)
 - Incidents, problems, and changes on discovered configuration items (CI).
 - Ownership tracking and assignment for mobile devices.

You can also do the following types of administrative actions:

- Device Management: You can locate, wipe, or retire a device. You can report on various aspects of the device.
- Integration with Azure Monitor: Delta notification.

Guided Setup

The guided setup for the Service Graph Connector for Microsoft Intune provides an organized sequence of tasks to configure the integration on your instance. To access the guided setup, see [Configure guided setup](#).

CMDB Integration Dashboards

The Integration Commons for CMDB store app provides a dashboard with a central view of the status, processing results, and processing errors of all installed integrations. You can see metrics for all integration runs. You can filter the view to a specific CMDB integration, a specific time duration, or a specific integration run. For more details about monitoring Microsoft Intune integrations in the CMDB Integrations Dashboard, see [Using the CMDB Integrations Dashboard](#).

Data Mappings

Data from data sources in the Microsoft Intune application is mapped and transformed into the ServiceNow CMDB Configuration Item (CI) class definitions using the Robust Transform Engine (RTE). Data is inserted into the ServiceNow CMDB using the Identification and Reconciliation Engine (IRE).

You can use the IntegrationHub ETL app to view the data maps. See [IntegrationHub ETL \(3.2\)](#) for more information.

The Microsoft Intune data sources include:

- SG-Intune Computer
- SG-Intune Devices
- SG-Intune Software

When you complete the guided setup, you can configure the integration to periodically pull data from the Microsoft Intune application.

The data is loaded into the following staging tables:

- SG-Intune Computer [sn_intune_integrat_computer]
- SG-Intune Devices [sn_intune_integrat_devices]
- SG-Intune Software [sn_intune_integrat_software]

The data is then inserted into the following target tables:

- Computer [cmdb_ci_computer]
- Handheld Computing Device [cmdb_ci_handheld_computing]
- IP Address [cmdb_ci_ip_address]
- Network Adapter [cmdb_ci_network_adapter]
- Serial Number [cmdb_serial_number]
- Software [cmdb_ci_spkg]
- Software Installation [cmdb_sam_sw_install]
- Software Instance [cmdb_software_instance]

Note:

- To view any additional information about a handheld device or computer, you can add the SG-Intune Device Related and SG-Intune Computer Related related lists by configuring the Related lists view.
- For any discovered software applications that were deleted later in the Microsoft Intune application, the Service Graph Connector automatically deletes the corresponding records in CMDB.

Set up authentication credentials and a scheduled job to import Microsoft Intune data into your CMDB.

Before you begin

To use this Service Graph Connector, you need a subscription to a Subscription Unit that is based in the IT Operations Management (ITOM) Visibility application or in the ITOM Discovery application. As defined in the section titled "Managed IT Resource Types" in [ServiceNow Subscription Unit Overview](#), for managed IT resources that are created or modified in the CMDB by this Service Graph Connector but that are not yet managed by [ITOM Visibility or ITOM Discovery](#), these resources will increase Subscription Unit consumption from that application. Review your current Subscription Unit consumption within ITOM Visibility or ITOM Discovery to ensure available capacity.

Before you start the configuration process, you need to grant Graph API permissions when the application is registered with Microsoft Intune. When you register the application, you will get credential information that will be needed to use the Microsoft Graph API to connect to the Microsoft Intune REST endpoints. For more information about how to configure the permissions, see the [Service Graph Connector for Microsoft Intune - Troubleshooting connection issues](#) blog post on ServiceNow Community.

Dependencies and requirements:

- The [Integration Commons for CMDB](#) store app, which is automatically installed.
- The [CMDB CI Class Models](#) store app store app, which is automatically installed.
- Datastream Action plugin (com.glide.hub.action_type.datastream), which is automatically installed.
- ITOM Licensing plugin (com.snc.itom.license). For more information, see [Request Discovery](#).

Starting with Service Graph Connector for Microsoft Intune version 2.1.1, a new feature introduces support for Multi-instance, which is the ability for SG-Intune to connect to and import data from multiple Microsoft Intune instances. This feature involves dynamically creating data sources and scheduled imports, thus requires the granting of additional permissions. Additional steps have been added in the guided setup to perform these steps.

Starting with the San Diego release, embedded help content will not be visible on the default Polaris theme in the Next Experience. You must activate the embedded help content by completing the following steps:

1. Select a task section in the guided setup, and then click **Configure**.
2. In the menu bar, click the help icon (?

Role required: admin

Procedure

1. Navigate to **All > Service Graph Connector for Microsoft Intune > Setup**.
2. On the Getting Started page, select **Get started**.
3. Configure the credentials.
 - a. On the Service Graph Connector for Microsoft Intune page, in the Configure the connection section, select the task **Configure credentials**.
 - b. On the next page, in the Configure credentials task section, click **Configure**.
 - c. On the form, fill in the fields.

Edit Connection form

Field	Description
Connection Name	Name of the Microsoft Intune application. This field is automatically set.
Connection URL	Connection URL for the connection. Based on the region of your Microsoft Intune application, enter the connection URL in one of the following formats: <ul style="list-style-type: none">• Global

Field	Description
	<p><code>https://graph.microsoft.com</code></p> <ul style="list-style-type: none"> • US Government <p><code>https://graph.microsoft.us</code></p> <ul style="list-style-type: none"> • China <p><code>https://microsoftgraph.chinacloudapi.cn</code></p> <ul style="list-style-type: none"> • Germany <p><code>https://graph.cloudapi.de/</code></p>
OAuth Client ID	The client ID of the Microsoft Intune application.
OAuth Client Secret	The client secret of the Microsoft Intune application.
OAuth Token URL	<p>Callback URL for the provider. Based on the region of your Microsoft Intune application, enter the token URL in one of the following formats:</p> <ul style="list-style-type: none"> • Global <p><code>https://login.microsoftonline.com/<tenantid>/oauth2/v2.0/token</code></p> <ul style="list-style-type: none"> • US Government <p><code>https://login.microsoftonline.us/<tenantid>/oauth2/v2.0/token</code></p>

Field	Description
	<p><code><tenantid>/oauth2/v2.0/token</code></p> <ul style="list-style-type: none"> • China <code>https://login.partner.microsoftonline.cn/<tenantid>/oauth2/v2.0/token</code> • Germany <code>https://login.microsoftonline.de/<tenantid>/oauth2/v2.0/token</code> <p>Where <code><tenantid></code> is the tenant ID of your Microsoft Intune application.</p>

- d. Click **Edit and Get OAuth Token**.
- e. Go back to the guided setup page and for the Configure credentials task, click **Mark as Complete**.
4. (Optional) If needed, configure the MID Server.
 - a. In the Configure MID Server section, select **Configure**.
 - b. Select the **Use MID server** check box.
 - c. Click **Update** to save the record.

Note: You do not need to update the other fields.
5. Test the connection to the Microsoft Intune API.
 - a. In the Test the connection section, select **Configure**.
 - b. On the form, review the fields.

Data Source form

Field	Description
Name	Unique name for this data source.
Import set table label	Label of the table that will be created for this data source.
Import set table name	Name of the table that will be created for this data source.
Data in single column	Option to set to data in single column.
Type	Data storage type of the data to be imported.
Application	Application containing this record.

- c. (Optional) Modify the properties in the Service Graph Connection Properties related list of the connection record.

Note: Try to retain the default value of the software_path property to ensure the proper retrieval of software details.

- d. Click the **Test Connection** related link to start the testing process.
e. When the **Status** field is set to **Success**, return to the guided setup page.
If any of the tests have errors, follow the suggestions for remediation.
f. In the Test the connection task section, click **Mark as Complete**.

6. Add multiple instances.

- a. On the left sidebar, click the Add Multiple Instances icon ().

- b. On the Service Graph Connector for Microsoft Intune page, in the Add Multiple Instances section, select the task **Update Data Source Access**.
 - c. In the Update Data Source Access section, select **Configure**.
 - d. To edit the record, select **Global** from the Scope menu.
 - e. Under the **Application Access** tab, select the **Can create**, **Can update**, and **Can delete** check boxes.
 - f. Save the record.
 - g. From the Scope menu, select **Service Graph Connector for Microsoft Intune**.
 - h. In the Help task bar, click **Mark as Complete**.
 - i. Repeat these steps in the Update Scheduled data import access section with the Scheduled data import [scheduled_data_set] table.
7. Clear the cache for the new connection.
- a. Select the **Clear Cache for Datasource and Import set** task then **Configure**.
 - b. Clear the cache by selecting **Global** from the Scope menu.
 - c. Enter the following script.

```
GlideTableManager.invalidateTable("sys_data_source");
GlideCacheManager.flushTable("sys_data_source");

GlideTableManager.invalidateTable("scheduled_import_set");
GlideCacheManager.flushTable("scheduled_import_set");

GlideTableManager.invalidateTable("sys_db_object");
GlideCacheManager.flushTable("sys_db_object");
```

- d. Select **Run Script**.
 - e. From the Scope menu, select **Service Graph Connector for Microsoft Intune**.
 - f. Click **Mark as Complete**.
8. To either add or save the connection, click **Configure** for the Create or Edit connection task.

- To add a connection, select **Add Connection**.
- To save the edits for the existing connection, select **Edit**.

Note: You need to the following information from your Microsoft Intune administrator:

- Client ID
- Client Secret
- Token URL

When a Microsoft Intune administrator registers an application, the Client ID, Client Secret, and Token URL will be available. To get more information about how to register an application, see the [Microsoft Intune documentation site](#).

- a. On the form, fill in the fields or edit as needed.

Create Connection form

Field	Description
Connection Name	Display name for the connection.
Connection URL	Connection URL for the new connection. Based on the region of your Microsoft Intune application, enter the connection URL in one of the following formats: <ul style="list-style-type: none">• Global

Field	Description
	<p><code>https://graph.microsoft.com</code></p> <ul style="list-style-type: none"> • US Government <p><code>https://graph.microsoft.us</code></p> <ul style="list-style-type: none"> • China <p><code>https://microsoftgraph.chinacloudapi.cn</code></p> <ul style="list-style-type: none"> • Germany <p><code>https://graph.cloudapi.de/</code></p>
OAuth Client ID	Client ID for the provider.
OAuth Client Secret	Client Secret for the provider.
OAuth Token URL	<p>Callback URL for the provider. Based on the region of your Microsoft Intune application, enter the token URL in one of the following formats:</p> <ul style="list-style-type: none"> • Global <p><code>https://login.microsoftonline.com/<tenantid>/oauth2/v2.0/token</code></p> <ul style="list-style-type: none"> • US Government <p><code>https://login.microsoftonline.us/<tenantid>/oauth2/v2.0/token</code></p> <ul style="list-style-type: none"> • China

Field	Description
	<p><code>https://login.partner.microsoftonline.cn/<tenantid>/oauth2/v2.0/token</code></p> <p>• Germany <code>https://login.microsoftonline.de/<tenantid>/oauth2/v2.0/token</code></p> <p>Where <tenantid> is the tenant ID of your Microsoft Intune application.</p>

- b. Either add or save the connection.
 - To create the new connection, select **Create and Get OAuth Tokens**.
 - To save the edits for the existing connection, select **Edit and Get OAuth Token**.
 - c. Navigate back to the guided setup and click **Mark as Complete**.
 - d. (Optional) Set up the MID Server for the connection you created.
 - a. In the Configure Mid Servers section, click **Configure**.
 - b. Select the name of the connection you created.
 - c. Click the **Use MID server** check box.
 - d. Click **Update**.
 - e. When you're finished with the task, click **Mark as Complete**
9. Configure the sets of the data sources and scheduled data imports for the new connection.
- a. In the Configure data sources and scheduled data imports section, select **Configure**.

- b. On the form, fill in the fields.

Field	Description
Instance Prefix to Data source and Scheduled data import sets	The prefix is an identifier that is used in all of the data source and scheduled import names for this distinct SG-Intune connection. In a multi-instance deployment, this prefix should be a short, meaningful identifier that allows you to identify a set of related data sources.
Connection and Credentials Alias	Select the connection alias that was created in the previous step.
Run Scheduled Import as User	Select a user to populate the field on the scheduled data import.

- c. Click **Submit** then **Mark as Complete**.
- d. Test the connection, in the Test New Connections section, by selecting **Configure**.
- Select the name of the data source associated with the newly created connection.
 - Click the **Test Connection** related link to start the process.
 - Optionally modify the properties in the Service Graph Connection Properties related list.

Microsoft Intune connection properties

Property	Description
api_version	The version of the Microsoft Intune Graph API.

Property	Description
software_path	<p>The path of the software code for finding apps and associated devices or vice versa. Leave the property value as is.</p>
include_primary_user_details	<p>Enable retrieving the details of the primary user during import and populating the assigned_to attribute in the records of the cmdb_ci table by setting the property value to true. For retrieving the enrolled user details, set the property value to false.</p> <p>Retrieving primary user details increases the time for importing data because of additional API calls.</p> <p>Note: When a user is assigned to a device initially, the enrolled and primary users are same. But if the device is reassigned to another user, the primary user name is reassigned to the new user, but the enrolled user is still the original enrolled user name.</p>
include_ip_address_details	<p>Enable retrieving the IP addresses of devices during import and populate the records in the cmdb_ci_ip_address table by setting the property value</p>

Property	Description
	<p>to true. To skip retrieving IP addresses, set the property to false.</p> <p>Retrieving IP addresses increases the time for importing data because of additional API calls.</p>

Note: The properties in the Service Graph Connection Properties related list of the connection record are modifiable. However, try to retain the default value of the software_path property to ensure the proper retrieval of software details.

- d. When the **Status** field is set to **Success**, return to the guided setup page.

Note: If any of the tests have errors, follow the suggestions for remediation.

- e. Return to the guided setup and click **Mark as Complete** for the Test New Connections task.

10. Set up the scheduled import jobs.

- a. On the left sidebar, click the Set up scheduled import jobs icon ().
- b. On the Service Graph Connector for Microsoft Intune page, in the Set up scheduled import jobs section, select the task **Configure the scheduled job**.
- c. In the Configure the scheduled import jobs task section, click **Configure**.
- d. Select the name of the scheduled import you want to run.
- e. On the form, fill in the fields.

Scheduled Data Import form

Field	Description
Name	Name of the scheduled job.
Data source	Data source record that defines the data to import.
Run as	Option to run the scheduled job with the credentials of the specified user.
Active	Option to activate the scheduled job. Select this option.
Concurrent Import	Function that loads the data from multiple import sets. The function then processes and transforms the data concurrently.
Partition Method	Partition method for the concurrent import set.
Partition Size	Import set size for early scheduling.
Execute pre-import script	Option to specify a script to run before the import is performed.
Execute post-import script	Option to specify a script to run after the import is performed.
Application	Application that contains this scheduled job.
Run	Frequency of running the import.

Field	Description
Conditional	Conditions under which this job is executed.

- f. Click **Execute Now** then **Mark as Complete**.

When you complete the guided setup, you can configure the integration to periodically pull data from Microsoft Intune. The data is saved in tables that extend from the Configuration item [cmdb_ci] table.

The following attributes in the Computer [cmdb_ci_computer] table are populated by collected data:

Attribute label	Attribute name
Name	name
Serial number	serial_number
Description	short_description
Disk space (GB)	disk_space
Manufacturer	manufacturer
Operating System	os
OS Version	os_version
Model ID	model_id
Assigned to	assigned_to
Chassis type	chassis_type

Relationships created for Computer

Parent class	Relationship type	Child class
Computer [cmdb_ci_computer]	Owns::Owned by	Network Adapter [cmdb_ci_network_adapter]
Computer [cmdb_ci_computer]	Owns::Owned by	IP Address [cmdb_ci_ip_address]
Computer [cmdb_ci_computer]	Reference	SG-Intune Computer Related [sn_intune_integrat_computer_related]
Computer [cmdb_ci_computer]	Reference	Software Installation [cmdb_sam_sw_install]

The following attributes in the Handheld Computing Device [cmdb_ci_handheld_computing] table are populated by collected data:

Attribute label	Attribute name
Name	name
Serial number	serial_number
Description	short_description
Disk space (GB)	disk_space
IMEI	imei
MEID	meid
Operating System	os
OS Version	os_version
Phone Number	phone_number

Attribute label	Attribute name
Root Access	root_access
Model ID	model_id
Carrier	carrier
Assigned to	assigned_to
Manufacturer	manufacturer

Relationships created for Handheld Computing Device

Parent class	Relationship type	Child class
Handheld Computing Device [cmdb_ci_handheld_computing]	Owns::Owned by	Network Adapter [cmdb_ci_network_adapter]
Handheld Computing Device [cmdb_ci_handheld_computing]	Owns::Owned by	IP Address [cmdb_ci_ip_address]
Handheld Computing Device [cmdb_ci_handheld_computing]	Reference	SG-Intune Device Related [sn_intune_integrat_device_related]
Handheld Computing Device [cmdb_ci_handheld_computing]	Reference	Software Installation [cmdb_sam_sw_install]

The following attributes in the Software Installation [cmdb_sam_sw_install] table are populated by collected data:

Attribute label	Attribute name
Display name	display_name

Attribute label	Attribute name
Version	version
Discovery source	discovery_source
Installed on	installed_on

Relationships created for Software Installation

Parent class	Relationship type	Child class
Software Installation [cmdb_sam_sw_install]	Reference	Handheld Computing Device [cmdb_ci_handheld_computing]

The following attributes in the Network Adapter [cmdb_ci_network_adapter] table are populated by collected data:

Attribute label	Attribute name
MAC Address	mac_address
Name	name
Configuration Item	cmdb_ci

Relationships created for Network Adapter

Parent class	Relationship type	Child class
Network Adapter [cmdb_ci_network_adapter]	Reference	Handheld Computing Device [cmdb_ci_handheld_computing]
Network Adapter [cmdb_ci_network_adapter]	Reference	Computer [cmdb_ci_computer]

The following attributes in the IP Address [cmdb_ci_ip_address] table are populated by collected data:

Attribute label	Attribute name
IP version	ip_version
Owned By Configuration Item	owned_by_cmdb_ci
IP Address	ip_address
Name	name

Relationships created for IP Address

Parent class	Relationship type	Child class
IP Address [cmdb_ci_ip_address]	Reference	Handheld Computing Device [cmdb_ci_handheld_computing]
IP Address [cmdb_ci_ip_address]	Reference	Computer [cmdb_ci_computer]

The following attributes in the Serial Number [cmdb_serial_number] table are populated by collected data:

Attribute label	Attribute name
Serial Number	serial_number
Serial Number Type	serial_number_type
Valid	valid

Relationships created for Serial Number

Parent class	Relationship type	Child class
Serial Number [cmdb_serial_number]	Reference	Handheld Computing Device [cmdb_ci_handheld_computing]

Parent class	Relationship type	Child class
Serial number [cmdb_serial_number]	Reference	Computer [cmdb_ci_computer]

The following attributes in the SG-Intune Computer Related [sn_intune_integrat_computer_related] table are populated by collected data:

Attribute label	Attribute name
Device ID	device_id
Azure AD Registered	azure_ad_registered
Compliance State	compliance_state
Device Enrollment Type	device_enrollment_type
Email Address	email_address
Encrypted	is_encrypted
Managed Device Owner Type	managed_device_owner_type
Management Agent	management_agent
Supervised	is_supervised

The following attributes in the SG-Intune Device Related [sn_intune_integrat_device_related] table are populated by collected data:

Attribute label	Attribute name
Device ID	device_id
Azure AD Registered	azure_ad_registered
Compliance State	compliance_state
Device Enrollment Type	device_enrollment_type

Attribute label	Attribute name
Email Address	email_address
Encrypted	is_encrypted
Managed Device Owner Type	managed_device_owner_type
Management Agent	management_agent
Supervised	is_supervised

The following attributes in the Software [cmdb_ci_spkg] table are populated by collected data:

Attribute label	Attribute name
Key	key
Name	name
Version	version

Relationship created for Software

Parent class	Relationship type	Child class
Software [cmdb_ci_spkg]	Reference	Software Instance [cmdb_software_instance]

The following attributes in the Software Instance [cmdb_software_instance] table are populated by collected data:

Attribute label	Attribute name
Name	name
Installed on	installed_on

Relationships created for Software Instance

Parent class	Relationship type	Child class
Software Instance [cmdb_software_instance]	Reference	Handheld Computing Device [cmdb_ci_handheld_computing]
Software Instance [cmdb_software_instance]	Reference	Computer [cmdb_ci_computer]

Service Graph Connector for Jamf (2.12.0)

Use the Service Graph Connector for Jamf to pull data from Jamf into your ServiceNow instance.

The Service Graph Connector for Jamf pulls data from computers, disks, networks, and software into the ServiceNow® Configuration Management Database (CMDB) application.

Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

Supported versions

- Supported versions:
 - Jamf API v2.1.0
 - Jamf Pro API 10.28.0
 - Jamf Pro 10.35.0
- Supported ServiceNow versions:
 - San Diego

- Tokyo
- Utah

Use Cases

The following are examples on how you can use the Service Graph Connector for different ServiceNow® applications:

- [IT Operations Management \(ITOM\) Visibility](#)
 - Detailed hardware and software inventory tracking for macOS hardware and apps. The tracking can be done with or without Software Asset Management (SAM).
 - Detailed hardware and software inventory tracking for hardware and apps for both iPhones and iPads. The tracking can be done with or without Software Asset Management (SAM).
 - Compliance tracking for mobile devices and end-user computers.
 - Duplicate data detection on imports to improve performance of nightly imports.
- [Software Asset Management \(SAM\) and IT Asset Management \(ITAM\)](#)
 - Software package and installation tracking.
 - License reclamation by detecting removed software.
 - Software Usage tracking.
- [IT Service Management \(ITSM\)](#)
 - Incidents, problems, and changes on discovered configuration items (CI).
 - Automatic ownership assignment based on top users.

Guided Setup

The guided setup for the Service Graph Connector for Jamf provides an organized sequence of tasks to configure the integration on your instance. To access the guided setup, see [Configure guided setup](#).

CMDB Integrations Dashboard

The Integration Commons for CMDB store app provides a dashboard with a central view of the status, processing results, and processing errors of all installed integrations. You can see metrics for all integration runs. You can filter the view to a specific CMDB integration, a specific time duration, or a specific integration run. For more details about monitoring Jamf integrations in the CMDB Integrations Dashboard, see [Using the CMDB Integrations Dashboard](#).

Data mapping

Data from the Jamf data source is mapped and transformed into the ServiceNow CMDB Configuration Item (CI) class definitions using the Robust Transform Engine (RTE). Data is inserted into the ServiceNow CMDB using the Identification and Reconciliation Engine (IRE).

When you complete the guided setup, you can configure the integration to periodically pull data from Jamf.

You can use the IntegrationHub ETL app to view the data maps. See [IntegrationHub ETL \(3.2\)](#) for more information.

The data is loaded into the following tables:

- SG-Jamf Computers [sn_jamf_integrate_sg_jamf_computers] staging table
- SG-Jamf Lookup Mac Software Bundle Ids [sn_cmdb_int_util_mac_software_bundleid_lookup] table. If you want to view the publisher information for applications installed on Mac devices, run this table before running SG-Jamf Computers or SG-Jamf Mobile Devices.
- SG-Jamf Mobile Devices [sn_jamf_integrate_sg_jamf_mobile_devices] table
- SG-Jamf Remove Computers Software [sn_jamf_integrate_jamf_remove_software] table

Note: Removes the imported computer software data that were later deleted from the source.

- SG-Jamf Remove Mobile Software
[sn_jamf_integrate_remove_mobile_software] table
 - Note:** Removes the imported mobile software data that were later deleted from the source.
- SG-Jamf Software Usage [sn_jamf_integrate_jamf_software_usage] table

The data is then inserted into the following target tables:

- CI Relationship [cmdb_rel_ci]
- Computer [cmdb_ci_computer]
- Disk [cmdb_ci_disk]
- Handheld Computing device [cmdb_ci_handheld_computing]
- IP Address [cmdb_ci_ip_address]
- Network Adapter [cmdb_ci_network_adapter]
- Printer [cmdb_ci_printer]
- Serial Number [cmdb_serial_number]
- Software [cmdb_ci_spkg]
- Software Installation [cmdb_sam_sw_install]
- Software Instance [cmdb_software_instance]
- Software Usage [samp_sw_usage]

Note: For the Computer [cmdb_ci_computer] and Handheld Computing device [cmdb_ci_handheld_computing] data sources, if you created multiple Jamf instances and want to know where the CIs originated from, you can identify the origins from the **Key** and **Value** columns. Additionally, you can view the Jamf Extension Attributes [sn_jamf_integrate_extension_attribute] and Most recent discovery [last_discovered] fields in both data sources. If you want to configure multiple instances, follow the steps in [Configure guided setup](#) in step 4.

Set up authentication credentials and scheduled jobs to import Jamf data into your CMDB.

Before you begin

To use this Service Graph Connector, you need a subscription to a Subscription Unit that is based in the IT Operations Management (ITOM) Visibility application or in the ITOM Discovery application. As defined in the section titled "Managed IT Resource Types" in [ServiceNow Subscription Unit Overview](#), for managed IT resources that are created or modified in the CMDB by this Service Graph Connector, but that are not yet managed by [ITOM Visibility or ITOM Discovery](#), these resources will increase Subscription Unit consumption from that application. Review your current Subscription Unit consumption within ITOM Visibility or ITOM Discovery to ensure available capacity.

Dependencies and requirements:

- The [Integration Commons for CMDB](#) store app, which is automatically installed.
- The [CMDB CI Class Models](#) store app store app, which is automatically installed.
- ITOM Licensing plugin (com.snc.itom.license). An unlicensed plugin that contains computation logic for SU consumption as necessary. For more information, see [Request Discovery](#).
- Jamf Classic API version 10.x.
- To access Jamf data: API user with read-only access to Jamf database.

Starting with the San Diego release, embedded help content will not be visible on the default Polaris theme in the Next Experience. You must activate the embedded help content by completing the following steps:

1. Select a task section in the guided setup, and then click **Configure**.
2. In the menu bar, click the help icon (?

Roles required: admin

Procedure

1. Navigate to **All > Service Graph Connectors > Jamf > Setup**.
2. On the Getting Started page, select **Get started**.
3. Configure the connection.
 - a. On the Service Graph Connector for Jamf page, in the Configure the properties section, select the task **Configure transform JSON max partial length**.
 - b. On the next page, in the Configure transform JSON max partial length section, select **Configure** and do the following:
 - a. Change the scope to **Global**.
 - b. To create a new system property, select **New**.
 - c. In the **Name** field, enter `com.glide.transform.json.max-partial-length`.
 - d. Set the **Value** field with `65536`.

Note: The Value field is required for the transform to work.

- e. Click **Mark as Complete**.
- c. Filter the personal applications.

Note: By default, the SG-Jamf imports all application records. To import only managed application records, set the `sn_jamf_integrate.import_managed_apps_only` property value to `true`.

- a. In the Filter personal applications section, select **Configure**.
- b. Select the name of the property.
- c. In the **Value** field, update the value from false to true.
- d. Click **Mark as Complete**.
- d. Enable the Jamf proAPI property.
 - a. In the Enable Jamf proAPI property section, click **Configure**.
 - b. Select the **Use Jamf Pro API** and **Jamf Pro 10.35 or higher** check box as needed.

Note: If you are using Jamf Pro 10.35 or higher, then you must select the **Jamf Pro 10.35** check box to avoid generating errors.
- c. Click **Update**.

Note: When you configure the Jamf HTTP connection, ensure that the **Base Path** field is cleared so that you can set up the endpoint URL for the **Jamf proAPI** property.
- d. Click **Mark as Complete**.
- e. Configure the Jamf authentication credentials.
 - a. In the Configure Jamf authentication credentials section, click **Configure**.
 - b. On the form, fill in the fields.

Basic Auth Credentials form

Field	Description
Name	Descriptive name of this authentication configuration.
User name	User name that is used to authenticate the HTTP request when this Basic

Field	Description
	<p>authentication profile is enabled.</p> <p>Note: The Jamf user must have a role with read privileges for Computers so that the integration can pull computer data.</p>
Password	<p>Password that is used to authenticate the HTTP request when this Basic authentication profile is enabled.</p>

Note: The Jamf user must have a role with read privileges for Computers to pull computer data and Mobile Devices so the integration can pull mobile device data.

- c. Click **Update** if necessary then **Mark as Complete**.
- f. Configure the Jamf HTTP connection.
 - a. In the Configure Jamf HTTP connection section, click **Configure**.
 - b. On the form, fill in the fields.

HTTP(s) Connection form

Field	Description
Name	Name of the connection.
Use MID server	Option to select a MID Server that sends this HTTP connection. Using a MID Server is not compatible with mutual authentication.

Field	Description
Host	<p>Target host value used by the connection. The Connection URL will automatically fill in the hostname.</p> <p>Note: Update the Host field with a Jamf base URL or IP address. For example, <code>demojamfhost.com</code> or <code>127.0.0.1</code>.</p>
Credential	Credential value used by this connection.
Connection alias	Connection value that is used to refer to the connection.
URL builder	URL builder that is used to build the connection URL.
Connection URL	Connection URL for the connection. You can either manually enter a URL or use the URL builder to build the connection string.
Mutual authentication	Option to set the connection with mutual authentication.
Protocol	<p>Underlying protocol used by the connection.</p> <p>Note: Update the Protocol field if you are using anything other than <code>https</code>.</p>

Field	Description
Active	Option to activate the HTTP connection.
Domain	Domain that contains the connection.
Override default port	Target value port that is used by the connection.
Base path	<p>Base path for HTTP(s) connection.</p> <p>Note: You do not need to update this field. This field is automatically set to / JSSResource. If you are upgrading the Service Graph Connector for Jamf and want to use the Jamf proAPI property, then you must clear this field.</p>

- c. Click **Update** if necessary then **Mark as Complete**.
- g. Test the connection.
 - a. In the Test the connection section, click **Configure**.
 - b. Review the fields on the form.

Data Source form

Field	Description
Name	Unique name of this data source.

Field	Description
Import set table label	Label of the import set table that this data source will produce.
Import set table name	Name of the table that will be created for this data source.
Type	Data storage type of the data to be imported.
Data in single column	Option to set the data in a single column.
Application	Application that contains this record.
Data Loader	Script that loads data in the import set table.

- c. Test the connection by clicking the **Test Load 20 Records** related link.

Testing the connection may take a few moments. The page is refreshed to show the test results.

Note: The connection is successful if the **HTTP Status** is **200**. If there is anything displayed in the **Error Code** and **Error Message** fields, then the connection failed and further troubleshooting is required. Do not click **Load All Records** during this setup.

- d. Click **Mark as Complete**.

4. Add multiple instances.

Note: If you do not need to add multiple instances, you can skip this step.

- a. On the left side bar, select the Add Multiple Instances icon ().
 - b. On the Service Graph Connector for Jamf page, under the Add Multiple Instances section, select the **Update Data Source Access** task.
 - c. On the next page, in the Update Data Source Access section, select **Configure**.
 - d. Select the Data Source [sys_data_source] table.
 - e. To edit the record, select **Global** from the Scope menu.
 - f. Under the **Application Access** tab, select the **Can create**, **Can update**, and **Can delete** check boxes.
 - g. Save the record.
 - h. From the Scope menu, select **Service Graph Connector for Jamf**.
 - i. In the Help task bar, click **Mark as Complete**.
 - j. Repeat these steps in the Update Scheduled data import access section with the Scheduled data import [scheduled_data_set] table.
5. Add another connection.

Note: Change the scope to **Service Graph Connector for Jamf**, otherwise you will be unable to load the additional connections.

- a. Under the Add Another Connection section, click **Configure**.
- b. In Flow Designer, select **Add Connection**.
- c. On the form, fill in the fields.

Connection form

Field	Description
Connection Name	Display name for the connection.

Field	Description
Connection URL	Connection URL for the new connection.
User name	User name credential for the new connection.
Password	Password credential for the new connection.

- d. Click **Create Connection**.
- e. Navigate back to the guided setup and click **Mark as Complete**.
- f. If needed, set up the MID Server for the connection you created.
 - a. In the Configure Mid Servers section, click **Configure**.
 - b. Select the name of the connection you created.
 - c. Click the **Use MID server** check box.
 - d. Click **Update**.
- e. When you're finished with the task, click **Mark as Complete**
- g. If needed, enable the **Jamf proAPI** property for the connection you created.
 - a. In the Enable Jamf proAPI property section, click **Configure**.
 - b. For the connection you created, under the **Use JAMF Pro API** column, change the value from **false** to **true**.
 - c. Click **Mark as Complete**.
- h. Test the new connections.
 - a. In the Test New Connections section, click **Configure**.
 - b. Select the name of the connection you want to test.
 - c. To validate the data source configuration, click the **Test Load 20 Records** button.

Note: If the test connection fails, there is an error in the Jamf connection that you must fix.

- d. When you're finished, click **Mark as Complete**.
6. Set up the scheduled import jobs.
 - a. On the left side bar, select the set up scheduled import jobs icon ().
 - b. On the Service Graph Connector for Jamf page, in the Set up scheduled import jobs section, select the task **Configure the scheduled job**.
 - c. In the Configure the scheduled job task section, click **Configure**.
 - d. Select the name of the scheduled job that you want to activate.
 - e. Review the pre-populated fields on the Scheduled Data Import form.

Note: By default, the SG-Jamf Mobile Devices scheduled job is automatically set as **Active** and runs SG-Jamf Computer as a parent.

Scheduled Data Import form

Field	Description
Name	Name of the scheduled job.
Data Source	Data source record that defines the data to import.
Run as	Option to run the scheduled job with the credentials of the specified user.
Active	Option to activate the scheduled job. Select this option.

Field	Description
Concurrent Import	Function that loads the data from multiple import sets. The function then processes and transforms the data concurrently.
Partition Method	Partition method for the concurrent import set.
Partition Size	Import set size for early scheduling.
Execute pre-import script	Option to specify a script to run before the import is performed.
Execute post-import script	Option to specify a script to run after the import is performed.
Application	Application that contains this scheduled job.
Run	Frequency of running the import.
Conditional	Conditions under which this job is executed.

- f. Click **Execute Now** and repeat these steps for the other imports if needed.
- g. In the Configure the scheduled job task section, click **Mark as Complete**.

When you complete the guided setup, you can configure the integration to periodically pull data from Jamf. The data is saved in tables that extend from the Configuration item [cmdb_ci] table.

The following attributes in the Computer [cmdb_ci_computer] table are populated by collected data:

Attribute label	Attribute name
MAC Address	mac_address
Name	name
Serial number	serial_number
CPU core count	cpu_core_count
CPU count	cpu_count
CPU name	cpu_name
CPU speed (MHz)	cpu_speed
CPU type	cpu_type
DNS Domain	dns_domain
Fully qualified domain name	fqdn
Most recent discovery	last_discovered
OS Service Pack	os_service_pack
OS Version	os_version
RAM (MB)	ram
Model ID	model_id
Assigned to	assigned_to
Manufacturer	manufacturer
Operating System	os

Relationships created for Computer

Parent class	Relationship type	Child class
Computer [cmdb_ci_computer]	Owns::Owned by	Network Adapter [cmdb_ci_network_adapter]
Computer [cmdb_ci_computer]	Owns::Owned by	IP Address [cmdb_ci_ip_address]
Computer [cmdb_ci_computer]	Contains::Contained by	Disk [cmdb_ci_disk]
Computer [cmdb_ci_computer]	Reference	Key Value [cmdb_key_value]
Computer [cmdb_ci_computer]	Reference	SG-Jamf Extension Attributes [sn_jamf_integrate_extension_attributes]

The following attributes in the Disk [cmdb_ci_disk] table are populated by collected data:

Attribute label	Attribute name
Manufacturer	manufacturer
Computer	computer
Model ID	model_id
Name	name
Serial number	serial_number
Most recent discovery	last_discovered
Size bytes	size_bytes

Relationship created for Disk

Parent class	Relationship type	Child class
Disk [cmdb_ci_disk]	Reference	Computer [cmdb_ci_computer]

The following attributes in the Handheld Computing Device [cmdb_ci_handheld_computing] table are populated by collected data:

Attribute label	Attribute name
Description	short_description
Carrier	carrier
Name	name
Serial number	serial_number
Disk space (GB)	disk_space
ICCID	iccid
IMEI	imei
MEID	meid
Operating System	os
OS Version	os_version
Phone Number	phone_number
Root Access	root_access
Manufacturer	manufacturer
Assigned to	assigned_to
Model ID	model_id

Relationships created for Handheld Computing Device

Parent class	Relationship type	Child class
Handheld Computing Device [cmdb_ci_handheld_computing]	Owns::Owned by	Network Adapter [cmdb_ci_network_adapter]
Handheld Computing Device [cmdb_ci_handheld_computing]	Owns::Owned by	IP Address [cmdb_ci_ip_address]
Handheld Computing Device [cmdb_ci_handheld_computing]	Reference	Key Value [cmdb_key_value]
Handheld Computing Device [cmdb_ci_handheld_computing]	Reference	SG-Jamf Extension Attributes [sn_jamf_integrate_extension_attributes]

The following attributes in the IP Address [cmdb_ci_ip_address] table are populated by collected data:

Attribute label	Attribute name
IP Address	ip_address
IP version	ip_version
Most recent discovery	last_discovered
Name	name
Nic	nic

Relationship created for IP Address

Parent class	Relationship type	Child class
IP Address [cmdb_ci_ip_address]	Reference	Network Adapter [cmdb_ci_network_adapter]

The following attributes in the Key Value [cmdb_key_value] table are populated by collected data:

Attribute label	Attribute name
Key	key
Value	value

The following attributes in the Network Adapter [cmdb_ci_network_adapter] table are populated by collected data:

Attribute label	Attribute name
MAC Address	mac_address
Name	name
Discovery Source	discovery_source
Most recent discovery	last_discovered

Relationships created for Network Adapter

Parent class	Relationship type	Child class
Network Adapter [cmdb_ci_network_adapter]	Reference	Computer [cmdb_ci_computer]
Network Adapter [cmdb_ci_network_adapter]	Reference	Handheld Computing Device [cmdb_ci_handheld_computing]

The following attributes in the Printer [cmdb_ci_printer] table are populated by collected data:

Attribute label	Attribute name
Name	name
IP Address	ip_address
Most recent discovery	last_discovered

The following attributes in the Serial Number [cmdb_serial_number] table are populated by collected data:

Attribute label	Attribute name
Serial Number	serial_number
Serial Number Type	serial_number_type
Valid	valid

Relationships created for Serial Number

Parent class	Relationship type	Child class
Serial Number [cmdb_serial_number]	Reference	Computer [cmdb_ci_computer]
Serial Number [cmdb_serial_number]	Reference	Handheld Computing Device [cmdb_ci_handheld_c omputing]

The following attributes in the SG-Jamf Extension Attributes [sn_jamf_integrate_extension_attributes] table are populated by collected data:

Attribute label	Attribute name
Id	id

Attribute label	Attribute name
Extension Attributes	extension_attributes

The following attributes in the Software [cmdb_ci_spkg] table are populated by collected data:

Attribute label	Attribute name
Version	version
Manufacturer	manufacturer
Key	key
Name	name

Relationship created for Software

Parent class	Relationship type	Child class
Software [cmdb_ci_spkg]	Reference	Software Instance [cmdb_software_instance]

The following attributes in the Software Instance [cmdb_software_instance] table are populated by collected data:

Attribute label	Attribute name
Name	name
Installed on	installed_on

Relationships created for Software Instance

Parent class	Relationship type	Child class
Software Instance [cmdb_software_instance]	Reference	Handheld Computing Device [cmdb_ci_handheld_computing]

Parent class	Relationship type	Child class
Software Instance [cmdb_software_instance]	Reference	Computer [cmdb_ci_computer]

Service Graph Connector for Microsoft SCCM (3.4.0)

Use the Service Graph Connector for Microsoft SCCM to pull data from Microsoft System Center Configuration Manager (SCCM) into your ServiceNow instance.

The Service Graph Connector for Microsoft SCCM (SG-SCCM) imports SCCM data into the ServiceNow® Configuration Management Database (CMDB) application. The integration does not write to the SCCM database and supports the Microsoft Endpoint Configuration Manager (MECM). The integration pulls data from computers, processors, operating systems, disks, networks, and software.

Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

Supported versions

- Supported Microsoft SCCM/MECM versions:
 - 2303
 - 2211
 - 2207
 - 2203
- Supported ServiceNow versions:
 - San Diego
 - Tokyo

- Utah
- Vancouver

Use Cases

The following ServiceNow applications have features that interact with the Service Graph Connector:

- [IT Operations Management \(ITOM\) Visibility](#)
 - Ability to get visibility into your infrastructure.
 - Detailed hardware and software inventory tracking. The tracking can be done with or without Software Asset Management (SAM).
 - Ability to detect delta changes for efficient incremental imports from SCCM to the Now Platform.
- [IT Service Management \(ITSM\)](#)
 - Incidents, problems, and changes on discovered configuration items (CI).
 - Automatic device ownership assignment.
- [Software Asset Management \(SAM\) and IT Asset Management \(ITAM\)](#)
 - Tight integration with Software Asset Management Professional and client software distribution workflows.
 - Inventory Software package and installation tracking.
 - Software Usage tracking.
 - License reclamation by detecting removed software.
 - Support for software editions, normalizing publisher information, and normalizing product Information.
 - Support for SCCM Asset Intelligence.

Guided Setup

The guided setup for the Service Graph Connector for Microsoft SCCM provides an organized sequence of tasks to configure the integration on your instance. To access the guided setup, see [Configure guided setup](#).

Note: When you install and configure the Service Graph Connector for Microsoft SCCM, only active SG-SCCM data sources will be configured with credentials or MID configuration during the guided setup. If a Software Asset Management plugin is enabled after the connector has been installed, any additional SAM-related data sources will not be configured with credentials and will not function correctly. In this scenario, in order for these new SAM-related data sources to function correctly, you must go back into the guided setup for SG-SCCM and re-enter the credentials and MID configuration.

CMDB Integrations Dashboard

The Integration Commons for CMDB store app provides a dashboard with a central view of the status, processing results, and processing errors of all installed integrations. You can see metrics for all integration runs. You can filter the view to a specific CMDB integration, a specific time duration, or a specific integration run. For more details about monitoring Microsoft SCCM integrations in the CMDB Integrations Dashboard, see [Using the CMDB Integrations Dashboard](#).

Data mapping

Data from the SCCM data sources is mapped and transformed into the ServiceNow CMDB Configuration Item (CI) class definitions using the Robust Transform Engine (RTE). Data is inserted into the ServiceNow® CMDB using the Identification and Reconciliation Engine (IRE).

You can use the IntegrationHub ETL app to view the data maps. See [IntegrationHub ETL \(3.2\)](#) for more information.

The SCCM data sources include the following:

- SG-SCCM Computer Identity

You can see the following data if it is available in SCCM:

- Asset Tag. If the Asset Tag data is available, the mapping can be optionally enabled via the guided setup.

- Assigned
- DNS Domain
- SG-SCCM Computer OU. This data source imports data about the Organizational Unit (OU) name. The data is stored in the SG-SCCM Computer Related [sn_sccm_integrate_sccm_2019_computer_related] table. To add the view, open the computer record, click the top header, and select **View > SG-SCCM Computer Related**.
- SG-SCCM Disk
- SG-SCCM Last Discovered Update. This data source will run at the end of the import schedules to update the last_discovered date on the Computer CI. The source will conduct a full pull of all the computers and bring in only the ResouceID and the LastHWSync columns to update the CMDB Computer table.
- SG-SCCM Network
- SG-SCCM Operating System
- SG-SCCM Processor
- SG-SCCM Removed Software. Ensure the **Use last run datetime** option is cleared for every run.
- SG-SCCM Removed Software AI. Ensure the **Use last run datetime** option is cleared for every run.
- SG-SCCM Software
- SG-SCCM Software AI

Note: If you have Asset Intelligence on your Microsoft SCCM instance, you have to run SG-SCCM Removed Software AI and SG-SCCM Software AI. If you do not have Asset Intelligence on your instance, you have to run SG-SCCM Removed Software and SG-SCCM Software.

When you complete the guided setup, you can configure the integration to periodically pull data from SCCM. The data is loaded into staging tables and then inserted into the following target tables:

- CI Relationship [cmdb_rel_ci]

- Computer [cmdb_ci_computer] (required)
- Disk [cmdb_ci_disk]
- IP address [cmdb_ci_ip_address]
- Network Adapter [cmdb_ci_network_adapter]
- Serial Number [cmdb_serial_number]
- Software [cmdb_ci_spkg]
- Software Installation [cmdb_sam_sw_install]
- Software Instance [cmdb_software_instance]
- Software Usage [samp_sw_usage]

By default, network adapters that are missing an IP address or MAC address are not imported. To include these network adapters in the import, do the following:

1. Navigate to **Service Graph Connector for Microsoft SCCM > Data Sources**.
2. Select **SG-SCCM Network**.
3. Remove the where clause from the **SQL statement** field.

Troubleshooting

For more troubleshooting information about Service Graph Connector for Microsoft SCCM, see the [Service Graph Connector for Microsoft SCCM - FAQ and Troubleshooting](#) blog post on the ServiceNow Community site.

Set up and validate data source connection credentials to import Microsoft SCCM data into your CMDB.

Before you begin

Important: If you are currently using a version of the Microsoft SCCM plugin, refer to the following topic, [Upgrade from the legacy SCCM plugin](#).

Note: If you have the Service Graph Connector for Microsoft SCCM version 2.1.6 or lower installed in your production environment, then you will need to contact Customer Service and Support for additional steps to remove the SCCM Discovery Source from being included when calculating the subscription unit consumption.

Confirm that you are in the SCCM application scope instead of a global scope so that you are able to save your information.

Dependencies and requirements:

- The [Integration Commons for CMDB](#) store app, which is automatically installed.
- The [CMDB CI Class Models](#) store app store app, which is automatically installed.
- Integration - JDBC (com.snc.integration.jdbc)
- Windows MID Server required for access to SCCM environment.

To access SCCM data, you must have appropriate access to the SCCM database. You must have sufficient credentials to query the SQL Server that contains the SCCM database. You must do the following:

1. Create a SQL Server account in order to connect to the SCCM database on the MID Server to use the data sources.
2. Connect to your SCCM SQL Server and configure the following:
 - a. Create a new login user name and password for SQL authentication.
 - b. Choose the SCCM database that ServiceNow data sources will connect to.
 - c. Assign the 'db_datareader' role membership to the new user.

Starting with Service Graph Connector for Microsoft SCCM version 3.0.4, a new feature introduces support for Multi-instance, which is the ability for SG-SCCM to connect to and import data from multiple Microsoft SCCM instances. This feature involves dynamically creating data sources and scheduled imports, thus requires granting of additional permissions. Additional steps have been added in the guided setup to perform these

steps. For more information about how to complete these steps, see [Service Graph Connector for Microsoft 3.0 Setup Guidelines \[KB1001248\]](#) in Now Support.

Starting with the San Diego release, embedded help content will not be visible on the default Polaris theme in the Next Experience. You must activate the embedded help content by completing the following steps:

1. Select a task section in the guided setup, and then click **Configure**.
2. In the menu bar, click the help icon (?

Role required: admin

Procedure

1. Navigate to **All > Service Graph Connector for Microsoft SCCM > Setup**.
2. On the Getting Started page, select **Get Started**.
3. On the Service Graph Connector for Microsoft SCCM page, in the Configure Data source and Scheduled data import access section, select the task **Configure Data source access**.
4. Create a data source and scheduled data import for a new connection in the Service Graph Connector for Microsoft SCCM.
 - a. In the Configure Data source access section, click **Configure**.
 - b. Edit the record by selecting **Global** from the Scope menu.
 - c. Under the **Application Access** tab, select the **Can create**, **Can update**, and **Can delete** check boxes.
 - d. Click **Update** to save the record.
 - e. From the scope menu, select **Service Graph Connector for Microsoft SCCM**.
 - f. In the Help task bar, click **Mark as Complete**.
 - g. Repeat these steps in the Update Scheduled data import access section with the Scheduled data import [scheduled_data_set] table.

5. Connect to multiple SCCM instances.

Note: If you want to configure a connection using integrated authentication, skip to step 6.

- a. On the left side bar, select the configure the connection icon ().
- b. On the Service Graph Connector for Microsoft SCCM page, under the Configure the connection section, select the **Configure connection** task.
- c. In the Configure connection section, select **Configure** to open the connection dashboard.
- d. In Flow Designer, select **Add Connection**.
- e. On the form, fill in the fields.

Create Connection form

Field	Description
Host	Host of the connection.
Database name	Database name for the new connection.
User name	User name credential for the new connection.
Password	Password credential for the new connection.

- f. Click **Create Connection**.
- g. Navigate back to the guided setup.
- h. Under the Configure connection section, select **Mark as Complete**.
6. Configure the connection using integrated authentication to the Microsoft SCCM database.

Note: Skip and mark this section complete if you already configured a connection in the previous step.

- a. In the Configure connection (Integrated authentication), select **Configure** to open the connection dashboard.

Note: Only use this step if the JDBC connection is using integrated authentication.

- b. In Flow Designer, select **Add Connection**.

- c. On the form, fill in the fields.

Create Connection form

Field	Description
Host	Host of the connection.
Database name	Database name for the new connection.

- d. Click **Create Connection**.

- e. Navigate back to the guided setup.

- f. Under the Configure connection (Integrated authentication) section, select **Mark as Complete**.

7. Configure the sets of the data sources and scheduled data imports for the new connection.

- a. On the left side bar, select Configure Data Sources and Scheduled Data Imports.

- b. On the Service Graph Connector for Microsoft SCCM page, under the Configure Data sources and Scheduled data imports section, select the **Configure Data Sources and Scheduled Data Imports** task.

- c. In the Configure Data Sources and Scheduled Data Imports section, select **Configure**.

- d. On the form, fill in the fields.

Field	Description
Prefix to Data source and Scheduled data import sets	<p>The prefix is an identifier that is used in all of the data source and scheduled import names for this distinct SCCM connection. In a multi-instance deployment, this prefix should be a short, meaningful identifier that allows you to identify a set of related data sources.</p> <p>Note: For example, assume you have two different SCCM instances for AMS and EMEA regions. You can use AMS and EMEA for easy identification and differentiation between the sets of data sources and schedules.</p>
Connection and Credentials Alias	Select the connection alias that was created in the previous step.
MID Server	<p>Select the MID Server for the connection.</p> <p>Note: It is mandatory to select a MID Server for an Integrated Authentication connection.</p>
Run Scheduled Import as User	Select a user to populate the field on the scheduled data import.

e. Click **Create/Update Imports**.

8. Validate the data sources.

- a. In the Validate Data Sources task section, click **Configure**.
- b. Review the fields on the Data Source form, which are automatically set.
Try not to update any values on this page. The data source is pre-configured to use the SCCM JDBC Connection that you set up in the previous step.

Data Source form

Field	Description
Name	Unique name of this data source.
Import set table label	Label of the table that will be created for this data source.
Import set table name	Name of the table that will be created for this data source.
Type	Data storage type of the data to be imported.
Use MID Server	MID server to use to access the JDBC server.
Format	Format of the data file.
Instance name	Named instance for SQLServer.
Database name	Name of the database.
Database port	Port of the database.
Application	Application that contains this record.
Use integrated authentication	Windows JDBC integrated authentication.

Field	Description
User name	User name for connecting to the JDBC server.
Password	Password for the JDBC server.
Server	Server name for the JDBC connection.
Query	Query type. Query all data from a table or run a specific SQL statement.
Query timeout	Number of seconds the JDBC driver will wait for a query to complete. Zero means no timeout. If timeout is exceeded, the integration considers the JDBC result inaccessible and places it in an Error state.
Connection timeout	Number of seconds before the MID server connection cache pool closes the connection and removes it. Zero means no timeout.
SQL statement	SQL statement to extract the desired data from the database.
Use last run datetime	<p>Option to control the amount of data that is retrieved from a database during an import run. If unselected, then all rows in the table specified are imported, every time.</p> <p>You might want to use this setting if this is a one-time import, or if all the data in</p>

Field	Description
	the target table is new. If selected, two additional fields appear, enabling you to select a datetime value to limit imported data to delta values only.

- c. Validate your data sources by clicking the **Test Load 20 Records** related link.

Note: If the displayed completion code is Success, then the sources are validated. But if the displayed completion code is Error, then there is an error in the SCCM JDBC Connection that you must fix.

- d. After the data sources have been validated, if you have the **Use Last RunDatetime** check box selected, then clear the **Last RunDatetime** field for the data source.
Cleaning the field avoids the risk of potential data loss due to the test load.

- e. In the Help sidebar, click **Back to Guided Setup**.

- f. In the Validate data sources task section, click **Mark as Complete**.

9. Configure the process for importing disk data when the Discovery application is running.

Note: To avoid creating duplicate records in the CMDB Disks [cmdb_ci_disk] table, you must configure the sn_sccm_integrate.sccm_disks_managed property and set its value to true. By default, the property value is set to false.

- a. For the Configure Disk Data Imports task, select **Configure**.
- b. On the System Property page, set the value of the sn_sccm_integrate.sccm_disks_managed property to true.
- c. Click **Update**.

- d. Mark the Configure Disk Data Imports task as complete by clicking **Mark as Complete**.

10. Configure the scheduled data imports.

Note: If the **Use Integrated Authentication** check box is selected for the data source, the run as user for the scheduled data import job must have the import_admin role. For more information, see the "["Use Integrated Authentication" is being unchecked when you run the scheduled import related to SG-SCCM or SG-SCCM Computer Identity Data Source \[KB1312810\]](#)" article in the Now Support Knowledge Base.

- a. On the left side bar, select the Set up Scheduled data imports icon ().
- b. On the Service Graph Connector for Microsoft SCCM page, in the Set up scheduled import jobs section, select the task **Configure scheduled jobs**.
- c. In the Configure scheduled jobs task section, click **Configure**. By default, the newly created SG-SCCM Computer Identity scheduled job with the prefix you named is inactive.
- d. On the Scheduled Data Import form, verify the field values for the scheduled job.
For more information, see [Schedule a data import](#).

All other SG-SCCM scheduled jobs that are active will run in their specified order after the Computer Identity scheduled job is finished running. You can modify the **Active** field for each scheduled job as appropriate for your configuration.

Important: If you are upgrading to the Service Graph Connector for Microsoft SCCM, deactivate the existing 'SG-SCCM Computer Identity scheduled data import'. Additionally, you cannot use the baseline scheduled data import or data sources that have a name starting with 'SG-SCCM' because they are used as templates for creating an instance of data sources and scheduled data imports.

- e. Click **Execute Now**.

- f. Repeat the steps 11 d and 11 e for each scheduled job.
- g. In the Help sidebar, click **Back to Guided Setup**.
- h. In the Configure Scheduled data imports task section, click **Mark as Complete**.
 - i. (Optional) If you are upgrading to the Service Graph Connector for Microsoft SCCM, then deactivate the existing 'SG-SCCM Computer Identity' scheduled data import.
In the Deactivate Legacy Scheduled Data Imports section, click **Configure**.
 - j. Select the 'SG-SCCM Computer Identity' scheduled data import and deselect the **Active** check box.
 - k. In the Deactivate Legacy Scheduled Data Imports task section, click **Mark as Complete**.
11. (Optional) Customize the Instance Data source SQL statement.
 - a. On the left side bar, select the Customize Instance Data source SQL statement icon ().
 - b. On the Service Graph Connector for Microsoft SCCM page, in Customize Instance Data source SQL statement section, select the task **Customizing Data source SQL statement**.
 - c. In the Customizing Data source SQL statement task section, click **Configure**.
 - d. Select the name of the instance in which you want to customize the SQL statement.
 - e. After you're finished, select **Update**.
 - f. In the Help task bar, click **Mark as Complete**.
12. Configure the mapping of the asset tag.
 - a. On the left side bar, select the Configure mapping for Asset tag icon ().

- b. On the Service Graph Connector for Microsoft SCCM page, in Configure mapping for Asset Tag section, select the task **Configure mapping for Asset Tag**.
- c. In the Configure mapping for Asset Tag section, click **Configure**.
- d. In the **Value** field, enter **true**.
- e. Click **Update**.
- f. In the Help task bar, click **Mark as Complete**.

Deactivate the scheduled imports and pull records so that you can upgrade from the Microsoft SCCM 2016 plugin or an earlier version of the plugin to the Service Graph Connector for Microsoft SCCM.

Before you begin

Role required: none

If you are upgrading to the Service Graph Connector for Microsoft SCCM, the following steps are mandatory. The steps must be executed in any instance with an existing install of the Microsoft SCCM 2016 plugin or with an earlier version of the plugin.

Note: The Service Graph Connector for Microsoft SCCM is an independent implementation that does not reuse any components from the Microsoft SCCM 2016 plugin.

About this task

The Service Graph Connector for Microsoft SCCM is a successor to the Microsoft SCCM 2016 plugin. If you already have the Microsoft SCCM 2016 plugin or an earlier version installed in your instance, you must follow these steps to enable the successful transition from using the SCCM plugin to using the Service Graph Connector. Additionally, you can run a SG-SCCM cleanup to delete the Network Adapters and Disks created from the older SCCM plugin to make the migration faster.

When the Service Graph Connector for Microsoft SCCM is installed in an instance, you must no longer run any components from the Microsoft SCCM 2016 plugin, including Data Sources and Scheduled Data Imports. The SCCM 2016 plugin Scheduled Data Imports must be tuned off. The

Service Graph Connector for Microsoft SCCM installs a new set of SG-SCCM Data Sources and Scheduled Data Imports.

Warning: This upgrade process should be performed and validated in a non-production or test instance that is based on a recent clone of the customer production instance. Failure to first validate upgrade in a non-production instance may result in unexpected outcomes and possible data loss or corruption.

Any customizations to the SCCM 2016 plugin will not automatically migrate. The customizations would have to be reimplemented in the Service Graph Connector for Microsoft SCCM, such as by using IntegrationHub-ETL.

Procedure

1. (Optional) If you are migrating from the SCCM 2016 plugin, then run the Migration Readiness Tool for Service Graph Connector for SCCM.

The plugin can be downloaded from the [ServiceNow Store](#).

The tool does not migrate any changes, but runs a series of ATF tests to identify any customizations that were done to the legacy SCCM 2016 plugin compared to the OOB SCCM 2016 plugin. It will not migrate the changes between the two plugins, but serves to alert any customizations.

2. After the tool has finished running, review the tests that were failed. A failed test means that a customization was made in the plugin. The customization could be a transform map. For example, if a transform map was modified, then it needs to be reimplemented in the connector.
3. Deactivate the scheduled imports from the older SCCM plugin. For more information on how to deactivate scheduled imports, see [Upgrade the SCCM integration version](#).
4. (Optional) Delete duplicate OS software records in the older SCCM plugin.

Note: There is no cleanup required for Computer OS Software records if SAM is enabled.

- a. Navigate to the Software Package or Software Instance table in the older SCCM plugin.
- b. Search for the duplicate OS records that you want to delete. You can tell which records are the SG-SCCM software OS records if the version number is included in the Name or Product Name.
- c. To delete the duplicate software OS records from the Software Package table, do the following:
 - a. On the Software Package table, select the duplicate OS record that does not have **SG-SCCM** in the **Discovery Source** column.
 - b. Delete the duplicate OS record and repeat with other records as needed.
- d. To delete the duplicate software OS records from the Software Instance table, do the following:
 - a. On the Software Instance table, select the duplicate OS record that has the **SCCM group ID** column empty.
 - b. Delete the duplicate OS record and repeat with other records as needed.

Note: This step is optional because there is a difference in the way software OS records are written to the CMDB, between the legacy SCCM plugin and Service Graph Connector. The SCCM plugin did not record values to sys_object_source, a discovery_source, or a sccm_group_id.

5. After you are finished deactivating the scheduled imports and deleting the duplicate OS records from the older SCCM plugin, configure the connector.
For instructions, see [Configure the Service Graph Connector for Microsoft SCCM](#).
6. (Optional) Clean up the Disk and Network Adapter records created by the SCCM plugin.
 - a. Confirm that you are in the Service Graph Connector for Microsoft SCCM application scope.
 - b. Navigate to **Scheduled Job > SG-SCCM CleanupUtil**.

- c. Select the SG-SCCM CleanupUtil scheduled job, and then switch to the global application scope.
- d. To make a copy of the scheduled job, right-click the header of the scheduled job and select **Insert and Stay**.
- e. Change the name to **SG-SCCM CleanupUtil Global**.
- f. Click **Update**.
- g. When you need to run the scheduled job, click **Execute Now**.
- h. (Optional) To check the progress of the run, do the following:
 - a. Navigate to **System Log > All**.
 - b. To filter the records for the script run, enter **SG-SCCM CleanupUtil** under the **Message** search box.
The script will have a log message for each batch so that you know the status of the current run and its progress.

Note: The duration of the run depends on the amount of Network Adapter and Disk data in the CMDB from the previous SCCM integration. The data must meet the condition for the cleanup.

What to do next

When you execute the Service Graph Connector components, your existing CMDB data created by the SCCM plugin becomes managed and maintained by Service Graph Connector for Microsoft SCCM.

Upgrade your Service Graph Connector for Microsoft SCCM. When you upgrade versions, you can select the files that you want to transfer from the older version to the newer version of the Service Graph Connector for Microsoft SCCM.

Before you begin

To avoid any serious upgrade issues, perform the upgrade in a test or development instance prior to upgrading in a production environment.

Role required: admin

About this task

When you upgrade from an older version to a newer version of the Service Graph Connector for Microsoft SCCM, there will always be Skipped Updates. The original SCCM data source files that were installed will have the last run date updated as part of the data source runs. The files will be treated as custom files by the platform. When the newer version is installed, the newer data sources will not be automatically upgraded.

Once you have upgraded to the newer version, you can check the upgrade status.

Procedure

1. In the left navigation bar, navigate to **System Diagnostics > Upgrade History**.
2. Click **Review Skipped Updates**.
3. Review all the updates and pick the data source files from the newer version.

Note: If you have customized the previous version, then pick the changes that you need from the older file and migrate them to the new file. Repeat this migration process for all the other skipped update files.

What to do next

After the upgrade, if there are changes to the SQL in the data source files or mapping, then do the following:

1. Clear the **Use last run datetime** check box in all the data sources.
2. Perform a full pull of data from all the data sources.

Note: Upgrading from Service Graph Connector for Microsoft SCCM v2.3 to v2.3.1 requires a minimum pull of Computer and Software, because the Source Native Key for the software within the SQL was changed in the SG-SCCM Removed Software data source. Without the correct Source Native Keys, it will not be able find the software in the CMDB.

If you have any conflicts, refer to the following documentation:

- [Revert a customization.](#)
- [Resolve a skipped update and set a resolution status.](#)
- [Skipped Changes to Review related list.](#)

Enable software editions so that you can gather edition information for products such as Adobe Acrobat, Microsoft SQL Server, and Windows Exchange Server into the Service Graph Connector for Microsoft SCCM.

Before you begin

Note: There are two types of setup that are required, one on the SCCM Manager and the other on the ServiceNow Instance. For more information on how to set up the SCCM Manager, see the SCCM Manager Setup section of the [Custom solution to gather editions in SCCM \[KB0721360\]](#) article on the HI Knowledge Base. When you're finished setting up the SCCM Manager, refer back to this task and complete the steps.

Role required: admin

About this task

You can set up the ServiceNow Instance on the Service Graph Connector for Microsoft SCCM.

Procedure

1. Navigate to **All > Service Graph Connector Microsoft SCCM > Import Schedules.**

2. Select the Software Edition scheduled import you created to edit this import record.
3. Select the **Active** check box.
4. Click **Update**.
5. Navigate to **Service Graph Connector Microsoft SCCM > Data Sources**.
6. Select the Software Edition data source you created.
7. Under the Transforms list, select **Update software install with edition**.
8. Edit the data source and select the **Active** check box.
9. Click **Update**.

What to do next

You can verify that the edition information has been gathered by doing the following:

1. Navigate to **Service Graph Connector Microsoft SCCM > Data Sources** and the Software Edition data source you want to verify.
2. Select **Load All Records**.

Note: If the displayed completion code is Success, then the software edition data source was executed successfully. If the displayed completion code is Error, then there is an error that must be fixed.

When you complete the guided setup, you can configure the integration to periodically pull data from Microsoft SCCM. The data is saved in tables that extend from the Configuration item [cmdb_ci] table.

The following attributes in the Computer [cmdb_ci_computer] table are populated by collected data:

Attribute label	Attribute name
Serial number	serial_number

Attribute label	Attribute name
CPU core count	cpu_core_count
CPU speed (MHz)	cpu_speed
CPU type	cpu_type
CPU manufacturer	cpu_manufacturer
Name	name
Asset tag	asset_tag
Assigned	assigned
Chassis type	chassis_type
Class	sys_class_name
CPU core thread	cpu_core_thread
CPU count	cpu_count
CPU name	cpu_name
Default Gateway	default_gateway
DNS Domain	dns_domain
Most recent discovery	last_discovered
Operating System	os
OS Address Width (bits)	os_address_width
OS Domain	os_domain
OS Service Pack	os_service_pack
OS Version	os_version

Attribute label	Attribute name
RAM (MB)	ram
Model ID	model_id
Manufacturer	manufacturer
Assigned to	assigned_to

Relationships created for Computer

Parent class	Relationship type	Child class
Computer [cmdb_ci_computer]	Owns::Owned by	Network Adapter [cmdb_ci_network_adapter]
Computer [cmdb_ci_computer]	Owns::Owned by	IP Address [cmdb_ci_ip_address]
Computer [cmdb_ci_computer]	Contains::Contained by	Disk [cmdb_ci_disk]
Computer [cmdb_ci_computer]	Reference	Key Value [cmdb_key_value]
Computer [cmdb_ci_computer]	Reference	SG-SCCM Computer Related [sn_sccm_integrate_sccm_2019_computer_related]

The following attributes in the Disk [cmdb_ci_disk] table are populated by collected data:

Attribute label	Attribute name
Disk space (GB)	disk_space
Size bytes	size_bytes

Attribute label	Attribute name
Computer	computer
Device ID	device_id
Name	name
Description	short_description
Device type	drive_type
Manufacturer	manufacturer
Model ID	model_id

Relationship created for Disk

Parent class	Relationship type	Child class
Disk [cmdb_ci_disk]	Reference	Computer [cmdb_ci_computer]

The following attributes in the IP Address [cmdb_ci_ip_address] table are populated by collected data:

Attribute label	Attribute name
IP Address	ip_address
IP version	ip_version
Name	name
Nic	nic

Relationship created for IP Address

Parent class	Relationship type	Child class
IP Address [cmdb_ci_ip_address]	Reference	Network Adapter [cmdb_ci_network_adapter]

The following attributes in the Key Value [cmdb_key_value] table are populated by collected data:

Attribute label	Attribute name
Key	key
Value	value

The following attributes in the Network Adapter [cmdb_ci_network_adapter] table are populated by collected data:

Attribute label	Attribute name
MAC Address	mac_address
Name	name
DHCP Enabled	dhcp_enabled
Netmask	netmask
Configuration Item	cmdb_ci

Relationship created for Network Adapter

Parent class	Relationship type	Child class
Network Adapter [cmdb_ci_network_adapter]	Reference	Computer [cmdb_ci_computer]

The following attributes in the Serial Number [cmdb_serial_number] table are populated by collected data:

Attribute label	Attribute name
Serial Number	serial_number
Serial Number Type	serial_number_type
Valid	valid

Relationship created for Serial Number

Parent class	Relationship type	Child class
Serial Number [cmdb_serial_number]	Reference	Computer [cmdb_ci_computer]

The following attributes in the SG-SCCM Computer Related [sn_sccm_integrate_sccm_2019_computer_related] table are populated by collected data:

Attribute label	Attribute name
Resource ID	resource_id
OU Name	ou_name

The following attributes in the Software [cmdb_ci_spkg] table are populated by collected data:

Attribute label	Attribute name
Key	key
Name	name
Version	version
Vendor	vendor
Manufacturer	manufacturer

Relationship created for Software

Parent class	Relationship type	Child class
Software [cmdb_ci_spkg]	Reference	Software Instance [cmdb_software_instance]

The following attributes in the Software Instance [cmdb_software_instance] table are populated by collected data:

Attribute label	Attribute name
Name	name
Installed on	installed_on
Install date	install_date
Sccm group ID	sccm_group_id
SCCM TimeStamp	sccm_timestamp

Relationship created for Software Instance

Parent class	Relationship type	Child class
Software Instance [cmdb_software_instance]	Reference	Computer [cmdb_ci_computer]

Service Graph Connector for OpenTelemetry (1.2.0)

Use the Service Graph Connector for OpenTelemetry to ingest Configuration Management Database (CMDB) data from the ServiceNow Cloud Observability (formerly Lightstep) application using REST APIs. Push events from the Cloud Observability application into ServiceNow with Event Management.

Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

Supported ServiceNow versions

- San Diego
- Tokyo
- Utah

Use cases

The following examples describe how you can use the Service Graph Connector for OpenTelemetry:

- Import project topologies from Cloud Observability so that the site reliability engineering (SRE) teams can have a single view for triage with other ServiceNow data points (such as a change request, an incident) from a single view.
- Import events created by Cloud Observability against monitored projects to pinpoint SRE triage activities.

Guided setup

The guided setup for the Service Graph Connector for OpenTelemetry provides an organized sequence of tasks to configure the integration on your instance. To access the guided setup, see [Configure guided setup](#).

CMDB integrations dashboard

The Integration Commons for CMDB store app provides a dashboard with a central view of the status, processing results, and processing errors of all installed integrations. You can see metrics for all integration runs. You can filter the view to a specific CMDB integration, a specific time duration, or a specific integration run. For more details about monitoring Cloud Observability integrations in the CMDB Integrations Dashboard, see [Using the CMDB Integrations Dashboard](#).

Data mapping

Data from the Cloud Observability data sources is mapped and transformed into the ServiceNow CMDB configuration item (CI) class definitions using the Robust Transform Engine (RTE). Data is inserted into the ServiceNow CMDB using the Identification and Reconciliation Engine (IRE).

When you complete the guided setup, you can configure the integration to pull data from the application periodically.

You can use the IntegrationHub ETL app to view the data maps. See [IntegrationHub ETL \(3.2\)](#) for more information.

The following data sources are included for the Cloud Observability application:

OpenTelemetry Resources

Imports all the Kubernetes clusters, workloads, and application services from the OpenTelemetry traces and loads the imported data in the OpenTelemetry Resources [sn_sg_lightstep_resources] staging table.

OpenTelemetry Pods

Imports all the Kubernetes pods from the containers and loads the imported data in the OpenTelemetry K8s Pods [sn_sg_lightstep_pods] staging table.

OpenTelemetry Containers

Imports all the containers from the OpenTelemetry traces and loads the imported data in the OpenTelemetry docker containers [sn_sg_lightstep_containers] staging table.

OpenTelemetry Container Images

Imports all the container images from the OpenTelemetry traces and loads the imported data in the OpenTelemetry container images [sn_sg_lightstep_container_images] staging table.

OpenTelemetry Services

Imports all the Kubernetes services from the OpenTelemetry traces and loads the imported data in the OpenTelemetry K8s Services [sn_sg_lightstep_kubernetes_services] staging table.

OpenTelemetry Dependency Map

Imports the dependency maps to get an aggregate view of the traced data and loads the imported data in the OpenTelemetry Dependency Map [sn_sg_lightstep_dependency_map] staging table. The data source also imports and loads any inferred services and their related services data in the Inferred service [sn_sg_lightstep_inferred_service] staging table.

Note: You need to link the imported inferred services with a CI manually. For more information, see [Inferred service linking](#).

The data from the staging tables is then inserted into the following target tables:

- Calculated Application Service [cmdb_ci_service_calculated]
- Docker Container [cmdb_ci_docker_container]
- Docker Image [cmdb_ci_docker_image]
- Key Value [cmdb_key_value]
- Kubernetes Cluster [cmdb_ci_kubernetes_cluster]
- Kubernetes Cronjob [cmdb_ci_kubernetes_cronjob]
- Kubernetes DaemonSet [cmdb_ci_kubernetes_daemonset]
- Kubernetes Deployment [cmdb_ci_kubernetes_deployment]
- Kubernetes Job [cmdb_ci_kubernetes_job]
- Kubernetes Namespace [cmdb_ci_kubernetes_namespace]
- Kubernetes Node [cmdb_ci_kubernetes_node]
- Kubernetes Pod [cmdb_ci_kubernetes_pod]
- Kubernetes ReplicaSet [cmdb_ci_kubernetes_replicaset]

- Kubernetes Service [cmdb_ci_kubernetes_service]
- Kubernetes StatefulSet [cmdb_ci_kubernetes_statefulset]
- Server [cmdb_ci_server]

For more information on where data is saved when pulling data from Cloud Observability, see [CMDB classes targeted](#).

Inferred service linking

The ServiceNow Cloud Observability application can infer the presence of an inferred service when the span calling the remote service has the necessary information. Service Graph Connector for OpenTelemetry provides the linking of inferred services and their related services with CIs in the CMDB. For more information, see [Linking inferred services](#).

Set up scheduled import jobs to pull in data from Service Graph Connector for OpenTelemetry (formerly Lightstep) into your Configuration Management Database (CMDB).

Before you begin

Dependencies and requirements:

- The [Integration Commons for CMDB](#) store app, which is automatically installed.
- The [CMDB CI Class Models](#) store app store app, which is automatically installed.
- The ITOM Discovery License plugin (com.snc.itom.discovery.license). You must activate this plugin.
- ITOM Licensing plugin (com.snc.itom.license). For more information, see [Request Discovery](#).
- The Datastream Action plugin (com.glide.hub.action_type.datastream), which is automatically installed.
- Observability Commons for CMDB (sn_observability), which is only required for event ingestion. For Event Management to work, the Observability Commons for CMDB app must be installed prior to

installing the connector. For more information, see [Observability Commons for CMDB](#) on the ServiceNow Store.

Note: If you have an earlier version of the Service Graph Connector for OpenTelemetry, then don't migrate data from the old connector. You must uninstall the previous version and run the new integration.

Role required: admin

Procedure

1. Ensure that the application scope is set to Service Graph Connector for OpenTelemetry by using the application picker.
For more information, see [Application picker](#).
 2. Navigate to **All > Service Graph Connectors > OpenTelemetry > Setup**.
 3. On the Getting started page, select **Get Started**.
 4. Configure the application properties to set up your organization and authentication credentials for sending requests to the ServiceNow Cloud Observability APIs.
 - a. In the Configure the connection section of the Service Graph Connector for OpenTelemetry page, select **Get started**.
 - b. For the Set up OpenTelemetry task, ensure that the following conditions are met in the Cloud Observability application.
 - a. You've signed up or identified an existing organization for your Cloud Observability account.
Confirm with your account management team if you're not sure whether an organization exists. To sign up for a free Cloud Observability account, contact your ServiceNow account representative.
 - b. The OpenTelemetry collectors and software development kit (SDKs) are deployed in your cloud or on-premises instances.
- These steps are mostly performed by a development, DevOps, or site reliability engineering (SRE) team.

- c. The OpenTelemetry collectors and SDKs are configured to send data to your Cloud Observability organization that you identified in step 4.b.i.

Important: Note down your projects, organization, and API key details of the Cloud Observability application to be used later during the installation of the connector.

For more information on API keys, OpenTelemetry, and Kubernetes configuration details, see the following topics on the Cloud Observability documentation site:

- [Create and manage API keys](#)
- [Where to begin](#)
- [Quick Start Kubernetes: Collector and Operator](#)
- [Quick Start: Tracing instrumentation](#)
- [Already using OpenTelemetry Collectors?](#)

Note: After you have set up the OpenTelemetry environment in the Cloud Observability application, return to the guided setup and mark the Set up OpenTelemetry task to complete by selecting **Mark as Complete**.

- c. Enter your Cloud Observability organization details from where you want to retrieve the projects, resources, and dependency-mapping information.
 - a. For the Set up your organization task, select **Configure**.
 - b. In the **Value** field of the Service Graph Connection Properties form, enter the name of your Cloud Observability organization.
 - c. Select **Update**.
 - d. Mark the Set up your organization task to complete by selecting **Mark as Complete**.
- d. Enter the API key details associated with the Cloud Observability application.

- a. For the Set up the API key task, select **Configure**.
 - b. In the **API Key** field of the API Key Credentials form, enter the API key associated with the Cloud Observability application that you noted down in step **4.b**.
 - c. Select **Update**.
 - d. Set the Set up the API key task to complete by selecting **Mark as Complete**.
 - e. Test the Cloud Observability API connection to import data from the Cloud Observability application.
 - a. For the Test the connection task, select **Configure**.
 - b. Select the **Test Connection** related link.
 - c. When the **Status** field is set to **Success**, select **X** to close the Test the connection dialog box and return to the guided setup page.
If any of the tests have errors, follow the suggestions for remediation.
 - d. Set the Test the connection task to complete by selecting **Mark as Complete**.
 - f. Retrieve all the projects included in your Cloud Observability organization.
 - a. For the Get projects task, select **Configure**.
 - b. Select **Get Projects**.
 - c. When the Project properties related list is populated with all the projects included in the organization you specified in step **4.c.ii**, select **X** to close the Get projects dialog box and return to the guided setup page.
 - d. Set the Get projects task to complete by selecting **Mark as Complete**.
5. (Optional) Configure additional configurations to set up lookback time, add excluded projects, run project diagnostics, and configure

integration settings for service maps and stale configuration items (CIs).

- a. In the Advanced settings section of the Service Graph Connector for OpenTelemetryService Graph Connector for Infoblox page, select **Continue**.
- b. Set up the lookback time from when you want to retrieve resources from the projects.

Lookback time is used to calculate the start and end time for retrieving resources from a project of an organization and should match the scheduled job frequency setting. For example, if the lookback time is set to 12 hours, resources are retrieved from the start time calculated as the current time minus 12 hours and until the end time that is the current time.

- a. For the Set up lookback time task, select **Configure**.
- b. In the **Value** field of the Service Graph Connection Properties form, enter the lookback time in hours.
- c. Select **Update**.
- d. Set the Set up lookback time task to complete by selecting **Mark as Complete**.
- e. Select the projects from which you don't want to retrieve resources.
 - a. For the Add excluded projects task, select **Configure**.
 - b. Select a project from the **Project** column.
 - c. Select the **Exclude project** check box.
 - d. Select **Update**.
- f. Repeat steps from 5.c.ii to 5.c.iv for each project that you want to exclude.
- g. Set the Add excluded projects task to complete by selecting **Mark as Complete**.

- d. Run project diagnostics to test the Resource API response and ensure that the API contains the Kubernetes cluster name, Kubernetes namespace, and Kubernetes nodes.
 - a. For the Run project diagnostics task, select **Configure**.
 - b. Select a project from the **Project** column.
 - c. Select **Run diagnostics**.
 - d. When the **Diagnostics status** field set to **Success**, select **X** to close the Run project diagnostics dialog box and return to the guided setup page.
If any of the Diagnostic tests have errors, follow the instructions in the **Diagnostics message** field to resolve the errors.
 - e. Repeat the steps **5.d.ii** to **5.d.iv** for each project that you want to run diagnostics.
 - f. Set the Run project diagnostics task to complete by selecting **Mark as Complete**.
- e. Configure the system properties for service maps and stale CIs.
 - a. For the Configure system properties for the connector task, select **Configure**.
 - b. Verify the default values for the properties or fill in the values for a custom configuration.
 - c. Select **Save**.
 - d. Set the Configure system properties for the connector task to complete by selecting **Mark as Complete**.
 6. Configure the scheduled jobs to import data from the Cloud Observability application.
 - a. In the Configure the scheduled import jobs section of the Service Graph Connector for OpenTelemetry page, select **Continue**.
 - b. For the Configure the scheduled job task, select **Configure**.
 - c. Select the scheduled job that you want to activate.

- d. On the Scheduled Data Import form, verify the field values for the scheduled job.
For more information, see [Schedule a data import](#).
 - e. Select **Execute Now**.
 - f. Repeat the steps [6.c](#) to [6.e](#) for each scheduled job for data import.
 - g. Select the back icon (<) to return to the guided setup page.
 - h. Set the Configure the scheduled job task to complete by selecting **Mark as Complete** in the guided setup.
7. Manage alerts and events by sending events from the Cloud Observability application to the ServiceNow Event Management application.
You can manage alerts and events only when the Observability Commons for CMDB application is installed.

Note: Enabling this feature requires a paid Cloud Observability license, but the feature is also available to free accounts for the purpose of evaluating the product in non-production environments. You can reach out to your ServiceNow account representative for more information.

- a. In the Manage alerts and events section of the Service Graph Connector for OpenTelemetry page, select **Continue**.
- b. Create a webhook destination for a project.
 - a. For the Create a webhook task, select **Configure**.
 - b. Select a project that is not excluded from the **Project** column.
 - c. Select **Create Webhook** to create a webhook destination in the Cloud Observability application.
 - d. After a success message appears, select **X** to close the Create a webhook dialog box and return to the guided setup page.

When the webhook is created successfully:

- The system automatically creates a user record for each webhook destination in your ServiceNow instance.
 - The user name of the user record starts with `ls_api_<project_name>` and the user is assigned the `evt_mgmt_integration` role.
- e. Repeat the steps [7.b.ii](#) to [7.b.iv](#) for each project for which you want to create a webhook.
 - f. Set the Create a webhook task to complete by selecting **Mark as Complete** in the guided setup.
- c. Configure the system property for ingesting events that don't have matching CIs in the CMDB.
 - a. For the Configure the property for unmatched CIs task, select **Configure**.
 - b. In the **Value** field, enter **true**.
 - c. Select **Update**.
 - d. Set the Configure the property for unmatched CIs task to complete by selecting **Mark as Complete** in the guided setup.

When you complete the guided setup, you can configure the integration to periodically pull data from ServiceNow Cloud Observability (formerly Lightstep). The data is saved in tables that extend from the Configuration item [cmdb_ci] table.

Calculated Application Service [cmdb_ci_service_calculated]

The following attributes in the Calculated Application Service [cmdb_ci_service_calculated] table are populated by collected data:

Attribute label	Attribute name
Service Populator	service_populator
Name	name
Hide from dashboard	hide_from_dashboard

Attribute label	Attribute name
Operational status	operational_status
Service Populator Status	populator_status
Service Type	type

Relationships created for Calculated Application Service

Parent class	Relationship type	Child class
Calculated Application Service [cmdb_ci_service_calculated]	Connects to::Connected by	Kubernetes Deployment [cmdb_ci_kubernetes_deployment]
Calculated Application Service [cmdb_ci_service_calculated]	Connects to::Connected by	Calculated Application Service [cmdb_ci_service_calculated]
Calculated Application Service [cmdb_ci_service_calculated]	Reference	Key Value [cmdb_key_value]

Docker Container [cmdb_ci_docker_container]

The following attributes in the Docker Container [cmdb_ci_docker_container] table are populated by collected data:

Attribute label	Attribute name
Container id	container_id
Install Status	install_status

Docker Image [cmdb_ci_docker_image]

The following attributes in the Docker Image [cmdb_ci_docker_image] table are populated by collected data:

Attribute label	Attribute name
Image id	image_id
Install Status	install_status
Name	name

Key Value [cmdb_key_value]

The following attributes in the Key Value [cmdb_key_value] table are populated by collected data:

Attribute label	Attribute name
Key	key
Value	value

Kubernetes Cluster [cmdb_ci_kubernetes_cluster]

The following attributes in the Kubernetes Cluster [cmdb_ci_kubernetes_cluster] table are populated by collected data:

Attribute label	Attribute name
Name	name
Namespace	namespace
Install Status	install_status

Relationships created for Kubernetes Cluster

Parent class	Relationship type	Child class
Kubernetes Cluster [cmdb_ci_kubernetes_cluster]	Cluster of::Cluster	Kubernetes Node [cmdb_ci_kubernetes_node]

Parent class	Relationship type	Child class
Kubernetes Cluster [cmdb_ci_kubernetes_cluster]	Contains::Contained by	Kubernetes Pod [cmdb_ci_kubernetes_pod]
Kubernetes Cluster [cmdb_ci_kubernetes_cluster]	Contains::Contained by	Kubernetes Namespace [cmdb_ci_kubernetes_namespace]
Kubernetes Cluster [cmdb_ci_kubernetes_cluster]	Contains::Contained by	Kubernetes Service [cmdb_ci_kubernetes_service]

Kubernetes Cronjob [cmdb_ci_kubernetes_cronjob]

The following attributes in the Kubernetes Cronjob [cmdb_ci_kubernetes_cronjob] table are populated by collected data:

Attribute label	Attribute name
Name	name
Namespace	namespace
Install Status	install_status

Relationships created for Kubernetes Cronjob

Parent class	Relationship type	Child class
Kubernetes Cronjob [cmdb_ci_kubernetes_cronjob]	Hosted on::Hosts	Kubernetes Cluster [cmdb_ci_kubernetes_cluster]
Kubernetes Cronjob [cmdb_ci_kubernetes_cronjob]	Reference	Kubernetes Cluster [cmdb_ci_kubernetes_cluster]

Kubernetes DaemonSet [cmdb_ci_kubernetes_daemonset]

The following attributes in the Kubernetes DaemonSet [cmdb_ci_kubernetes_daemonset] table are populated by collected data:

Attribute label	Attribute name
Kubernetes Cluster	cluster
Name	name
Namespace	namespace
Install Status	install_status

Relationships created for Kubernetes DaemonSet

Parent class	Relationship type	Child class
Kubernetes DaemonSet [cmdb_ci_kubernetes_daemonset]	Hosted on::Hosts	Kubernetes Cluster [cmdb_ci_kubernetes_cluster]
Kubernetes DaemonSet [cmdb_ci_kubernetes_daemonset]	Reference	Kubernetes Cluster [cmdb_ci_kubernetes_cluster]

Kubernetes Deployment [cmdb_ci_kubernetes_deployment]

The following attributes in the Kubernetes Deployment [cmdb_ci_kubernetes_deployment] table are populated by collected data:

Attribute label	Attribute name
Kubernetes Cluster	cluster
Name	name

Attribute label	Attribute name
Namespace	namespace
Install Status	install_status

Relationships created for Kubernetes Deployment

Parent class	Relationship type	Child class
Kubernetes Deployment [cmdb_ci_kubernetes_deployment]	Hosted on::Hosts	Kubernetes Cluster [cmdb_ci_kubernetes_cluster]
Kubernetes Deployment [cmdb_ci_kubernetes_deployment]	Connects to::Connected by	Kubernetes Node [cmdb_ci_kubernetes_node]
Kubernetes Deployment [cmdb_ci_kubernetes_deployment]	Connects to::Connected by	Kubernetes ReplicaSet [cmdb_ci_kubernetes_replicaset]
Kubernetes Deployment [cmdb_ci_kubernetes_deployment]	Reference	Kubernetes Cluster [cmdb_ci_kubernetes_cluster]

Kubernetes Job [cmdb_ci_kubernetes_job]

The following attributes in the Kubernetes Job [cmdb_ci_kubernetes_job] table are populated by collected data:

Attribute label	Attribute name
Kubernetes Cluster	cluster
Kubernetes UID	k8s_uid
Name	name

Attribute label	Attribute name
Namespace	namespace
Install Status	install_status

Relationships created for Kubernetes Job

Parent class	Relationship type	Child class
Kubernetes Job [cmdb_ci_kubernetes_job]	Hosted on::Hosts	Kubernetes Cluster [cmdb_ci_kubernetes_cluster]
Kubernetes Job [cmdb_ci_kubernetes_job]	Reference	Kubernetes Cluster [cmdb_ci_kubernetes_cluster]

Kubernetes Namespace [cmdb_ci_kubernetes_namespace]

The following attributes in the Kubernetes Namespace [cmdb_ci_kubernetes_namespace] table are populated by collected data:

Attribute label	Attribute name
Kubernetes Cluster	cluster
Name	name
Install Status	install_status

Relationship created for Kubernetes Namespace

Parent class	Relationship type	Child class
Kubernetes Namespace [cmdb_ci_kubernetes_namespace]	Reference	Kubernetes Cluster [cmdb_ci_kubernetes_cluster]

Kubernetes Node [cmdb_ci_kubernetes_node]

The following attributes in the Kubernetes Node [cmdb_ci_kubernetes_node] table are populated by collected data:

Attribute label	Attribute name
Kubernetes UID	k8s_uid
Name	name
Install Status	install_status
Namespace	namespace
Kubernetes Cluster	cluster

Relationships created for Kubernetes Node

Parent class	Relationship type	Child class
Kubernetes Node [cmdb_ci_kubernetes_node]	Hosted on::Hosts	Server [cmdb_ci_server]
Kubernetes Node [cmdb_ci_kubernetes_node]	Reference	Kubernetes Cluster [cmdb_ci_kubernetes_cluster]

Kubernetes Pod [cmdb_ci_kubernetes_pod]

The following attributes in the Kubernetes Pod [cmdb_ci_kubernetes_pod] table are populated by collected data:

Attribute label	Attribute name
Name	name
Install Status	install_status
Kubernetes Cluster	cluster

Attribute label	Attribute name
Kubernetes UID	k8s_uid
Namespace	namespace
IP Address	ip_address
Start date	start_date

Relationships created for Kubernetes Pod

Parent class	Relationship type	Child class
Kubernetes Pod [cmdb_ci_kubernetes_pod]	Contains::Contained by	Docker Image [cmdb_ci_docker_image]
Kubernetes Pod [cmdb_ci_kubernetes_pod]	Contains::Contained by	Docker Container [cmdb_ci_docker_container]
Kubernetes Pod [cmdb_ci_kubernetes_pod]	Reference	Kubernetes Cluster [cmdb_ci_kubernetes_cluster]

Kubernetes ReplicaSet [cmdb_ci_kubernetes_replicaset]

The following attributes in the Kubernetes ReplicaSet [cmdb_ci_kubernetes_replicaset] table are populated by collected data:

Attribute label	Attribute name
Kubernetes UID	k8s_uid
Name	name
Namespace	namespace
Install Status	install_status

Attribute label	Attribute name
Kubernetes Cluster	cluster

Relationships created for Kubernetes ReplicaSet

Parent class	Relationship type	Child class
Kubernetes ReplicaSet [cmdb_ci_kubernetes_replicaset]	Hosted on::Hosts	Kubernetes Cluster [cmdb_ci_kubernetes_cluster]
Kubernetes ReplicaSet [cmdb_ci_kubernetes_replicaset]	Connects to::Connected by	Kubernetes Node [cmdb_ci_kubernetes_node]
Kubernetes ReplicaSet [cmdb_ci_kubernetes_replicaset]	Reference	Kubernetes Cluster [cmdb_ci_kubernetes_cluster]

Kubernetes Service [cmdb_ci_kubernetes_service]

The following attributes in the Kubernetes Service [cmdb_ci_kubernetes_service] table are populated by collected data:

Attribute label	Attribute name
Name	name
Namespace	namespace
Install Status	install_status

Kubernetes StatefulSet [cmdb_ci_kubernetes_statefulset]

The following attributes in the Kubernetes StatefulSet [cmdb_ci_kubernetes_statefulset] table are populated by collected data:

Attribute label	Attribute name
Name	name

Attribute label	Attribute name
Kubernetes Cluster	cluster
Namespace	namespace
Install Status	install_status

Relationship created for Kubernetes StatefulSet

Parent class	Relationship type	Child class
Kubernetes StatefulSet [cmdb_ci_kubernetes_statefulset]	Hosted on::Hosts	Kubernetes Cluster [cmdb_ci_kubernetes_cluster]
Kubernetes StatefulSet [cmdb_ci_kubernetes_statefulset]	Reference	Kubernetes Cluster [cmdb_ci_kubernetes_cluster]

Server [cmdb_ci_server]

The following attribute in the Server [cmdb_ci_server] table is populated by collected data:

Attribute label	Attribute name
Name	name

Related concepts

- Kubernetes extension classes
- CI relationships in the CMDB

Create inferred services relationships in your ServiceNow instance with other Cloud Observability application services as originally configured in the application by linking an inferred service configuration item (CI) in CMDB with an inferred service from the Cloud Observability application.

Inferred services

The inferred services in the Cloud Observability application are referred to as external services, libraries, or dependencies such as a database or a third-party API that haven't been instrumented with OpenTelemetry. These types of technologies are manually defined in the Cloud Observability application, and you can import them into your ServiceNow instance through the Service Graph Connector for OpenTelemetry.

For information about adding inferred services in the Cloud Observability application, see [Add inferred services](#) in the Cloud Observability documentation.

Storing inferred services data

The OpenTelemetry Dependency Map [sn_sg_lightstep_dependency_map] data source available with the Service Graph Connector for OpenTelemetry iterates through all the included projects in a Cloud Observability organization and pulls in the inferred services details. Inferred services details include mapping of related services. The data source saves these details in the Inferred service [sn_sg_lightstep_inferred_service] table.

The following attributes in the Inferred service [sn_sg_lightstep_inferred_service] table are populated by the OpenTelemetry Dependency Map [sn_sg_lightstep_dependency_map] data source.

Attribute label	Attribute name
Project	project
Organization	organization
ID	id
Inferred service	inferred_service
Inferred service CI	inferred_service_ci
Active	active

Attribute label	Attribute name
Last scan	last_scan
Cloud Observability CI	cloud_observability_ci

Due to insufficient information for defining identification rules, the pulled in inferred services don't automatically bind to a CI in CMDB. However, as a user with the cmdb_inst_admin role, you can manually link a CI to an inferred service. The linking creates appropriate relationships with the related services and generates the inferred service mapping in the respective application service maps of your ServiceNow instance.

Link an inferred service

Link a Cloud Observability inferred service with a CI to create relationships between the inferred service and other Cloud Observability services in your ServiceNow instance.

Before you begin

Role required: cmdb_inst_admin

Procedure

1. Navigate to **All > Service Graph Connectors > OpenTelemetry > Inferred services**.
2. Review the inferred services available from the Cloud Observability app in the **Inferred service** column.
3. For an inferred service, double-click the **Inferred service CI** column cell.
4. Select the lookup using list icon () to search for and select an inferred service CI available within the Configuration item [cmdb_ci] table.
5. Select the save icon ()
6. Repeat the steps 3 to 5 for each inferred service that you want to link with a CI.

Result

The Update CI relationship business rule is triggered that creates appropriate relationships for the inferred service with the related services and generates the inferred service mapping in the respective application service maps.

Note:

- If you remove any mapping later, the Delete CI relationship business rule is triggered to delete any relationships between the inferred service CIs and the Cloud Observability inferred services.
- For any inferred services that were not last scanned in the Cloud Observability app, the Service Graph Connector automatically deactivates the corresponding records in CMDB and deletes the relationship for the inactive records.

Service Graph Connector for SolarWinds (2.4.1)

Use the Service Graph Connector for SolarWinds to pull in data from the SolarWinds software into your ServiceNow instance.

The Service Graph Connector for SolarWinds pulls in asset inventory data (hardware and software) from the SolarWinds database into the ServiceNow® Configuration Management Database (CMDB) application.

Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

Supported versions

- Supported SolarWinds Orion minimum versions:
 - 2019.4 HF6
 - 2020.2.1. HF2

- Supported ServiceNow versions:
 - Quebec
 - Rome
 - San Diego
 - Tokyo

Use cases

The following are examples on how you can use the Service Graph Connector:

- Automatic normalization of asset information for hardware, virtualization and cloud resources, and software.
- Ability to configure and save synchronization schedules.

Guided setup

The guided setup for the Service Graph Connector for SolarWinds provides an organized sequence of tasks to configure the integration on your instance. To access the guided setup, see [Configure guided setup](#).

CMDB Integrations Dashboard

The Integration Commons for CMDB store app provides a dashboard with a central view of the status, processing results, and processing errors of all installed Service Graph Connectors. You can see metrics for all integration runs. You can filter the view to a specific integration, a specific time duration, or a specific integration run. For more details about monitoring SolarWinds integrations in the CMDB Integrations Dashboard, see [Using the CMDB Integrations Dashboard](#).

Data mappings

Data from data sources in the SolarWinds software is mapped and transformed into ServiceNow CMDB tables using the Robust Transform Engine (RTE). Data is inserted into ServiceNow CMDB using the Identification and Reconciliation Engine (IRE).

When you complete the guided setup, you can configure the integration to periodically pull data from the SolarWinds software.

You can use the IntegrationHub ETL app to view the data maps. See [IntegrationHub ETL \(3.2\)](#) for more information.

The data is loaded into staging tables and then inserted into the following CMDB target tables:

- AIX Server [cmdb_ci_aix_server]
- Availability Zone [cmdb_ci_availability_zone]
- CI Relationship [cmdb_rel_ci]
- Cloud Network [cmdb_ci_network]
- Cloud Service Account [cmdb_ci_cloud_service_account]
- Cloud Subnet [cmdb_ci_cloud_subnet]
- Computer [cmdb_ci_computer]
- Disk [cmdb_ci_disk]
- Hardware [cmdb_ci_hardware]
- Hardware Type [cmdb_ci_compute_template]
- Hyper-V Server [cmdb_ci_hyper_v_server]
- IIS Virtual Directory [cmdb_ci_iisdirectory]
- Image [cmdb_ci_os_template]
- IP Address [cmdb_ci_ip_address]
- Linux Server [cmdb_ci_linux_server]
- Logical Datacenter [cmdb_ci_logical_datacenter]
- Microsoft iis Web Server [cmdb_ci_microsoft_iis_web_server]
- MS SQL Server [cmdb_ci_db_mssql_server]
- MS SQL DataBase [cmdb_ci_db_mssql_database]
- Network Adapter [cmdb_ci_network_adapter]
- Network Gear [cmdb_ci_netgear]

- Serial Number [cmdb_serial_number]
- Server [cmdb_ci_server]
- Software [cmdb_ci_spkg]
- Software Installation [cmdb_sam_sw_install]
- Software Instance [cmdb_software_instance]
- Solaris Server [cmdb_ci_solaris_server]
- Storage Volume [cmdb_ci_storage_volume]
- UNIX Server [cmdb_ci_unix_server]
- VM Instance [cmdb_ci_vm_instance]
- Windows Server [cmdb_ci_win_server]

Set up authentication credentials and scheduled jobs to import SolarWinds data into your CMDB.

Before you begin

To use this Service Graph Connector, you need a subscription to a Subscription Unit that is based in the IT Operations Management (ITOM) Visibility application or in the ITOM Discovery application. As defined in the section titled "Managed IT Resource Types" in [ServiceNow Subscription Unit Overview](#), for managed IT resources that are created or modified in the CMDB by this Service Graph Connector, but that are not yet managed by [ITOM Visibility](#) or [ITOM Discovery](#), these resources will increase Subscription Unit consumption from that application. Review your current Subscription Unit consumption within ITOM Visibility or ITOM Discovery to ensure available capacity.

Before you start the configuration, navigate to [System Definition > Business Rules](#) and deactivate the ValidateServiceAccountId business rule.

Dependencies and requirements:

- The [Integration Commons for CMDB](#) store app, which is automatically installed.

- The [CMDB CI Class Models store app](#) store app, which is automatically installed.
- Discovery Core plugin (com.snc.discovery.core), which is automatically installed by Discovery.
- ITOM Discovery License plugin (com.snc.item.discovery.license). You must activate this plugin.
- ITOM Licensing plugin (com.snc.item.license). For more information, see [Request Discovery](#).
- SolarWinds Orion Platform.
- SolarWinds Server & Application Monitor and/or Network Performance Monitor.

Starting with the San Diego release, embedded help content will not be visible on the default Polaris theme in the Next Experience. You must activate the embedded help content by completing the following steps:

1. Select a task section in the guided setup, and then click **Configure**.
2. In the menu bar, click the help icon (ⓘ).

Roles required:

- To configure the ServiceNow platform: admin
- To access SolarWinds data: User with SWIS (SolarWinds Information Service) access (the same as through the Orion website, not database users).

About this task

Note: SolarWinds Integration can work with or without Service & Application Monitor (SAM) and Network Performance Monitor (NPM) being installed.

Procedure

1. Navigate to **All > Service Graph Connectors > SolarWinds > Setup**.
2. On the Getting started page, click **Get Started**.

3. Configure your authentication credentials used to connect to the Solarwinds SWIS API.
 - a. On the Service Graph Connector for SolarWinds page, in the Configure the connection section, select the task **Configure the authentication credentials used to connect to SolarWinds SWIS API.**
 - b. On the next page, in the Configure the authentication credentials task section, click **Configure**.
 - c. On the form, fill in the fields.

Basic Auth Credentials form

Field	Description
Name	Human-readable name for this credential. This field is automatically set. You can optionally change this setting.
User name	SolarWinds user name. Note: The SolarWinds user must have a role with read privileges to the SolarWinds Information Service API for all SolarWinds data sources that the integration can pull data.
Password	SolarWinds password which is stored in the database in encrypted form.
Active	Option to check if this credential is active.
Credential alias	Advanced selection criteria for this credential.

Field	Description
Order	Order in which credentials are tried. Smaller numbers are tried first.

- d. Click **Update** if necessary.
 - e. In the Configure the authentication credentials used to connect to the SolarWinds SWIS API task section, click **Mark as Complete**.
4. Configure the SolarWinds HTTP connection.
- a. In the Configure SolarWinds HTTP connection task section, click **Configure**.
 - b. Review the HTTP(s) Connection form and fill in fields as needed.

HTTP(s) Connection form

Field	Description
Name	Name of the connection. This field is automatically set. You can optionally change.
Use MID server	Option to enable this Connection to use MID server or not.
Host	Target host value used by the connection. This field is automatically set by the connection URL.
Credential	Credential value used by this connection.
Connection alias	Connection alias value with which the connection can be referred.

Field	Description
URL builder	URL builder that is used to build the connection URL.
Mutual authentication	Option to enable Mutual Authentication.
Protocol	Underlying protocol used by the connection.
Active	Option to activate the HTTP connection.
Domain	Domain to which the connection belongs.
Override default port	Target port value used by Connection.
Base path	Base path for HTTP(s) connection that is required but should not be modified.

- c. Click **Update** if necessary.
 - d. In the Configure SolarWinds HTTP connection task section, click **Mark as Complete**.
5. Configure the SolarWinds modules.
- a. In the Configure SolarWinds Modules section, click **Configure**.
 - b. Update the **Value** for the sn_solarwinds_inte.npm_installed and the sn_solarwinds_inte.sam_installed properties in one of the following ways:
 - If the respective module is installed on the SolarWinds instance, then set the fields **true**.
 - If the respective module is not installed on the SolarWinds instance, then set the fields to **false**.

Note: This update changes the API class and the data that is returned from them.

- c. In the Help sidebar, click **Mark as Complete**.
6. Test the connection.
 - a. In the Test the connection task section, click **Configure**.
 - b. Review the fields on the Data Source form, which are automatically set.

Data Source form

Field	Description
Name	Unique name for this data source.
Import set table label	Label of the table that will be created for this data source.
Import set table name	Name of the table that will be created for this data source.
Type	Data storage type of the data to be imported.
Data in single column	Data in single column.
Application	Application containing this record.
Data stream action	The Data Source request action that will be invoked to get data.

- c. Test the connection by clicking the **Test Load 20 Records** related link.

Testing the connection takes a few moments, after which the page refreshes to show the test results.

This step tests the SG-Solarwinds Hardware data source and ensures that data is loaded into the staging table. A successful connection for SG-Solarwinds Hardware means that all SolarWinds data sources connect successfully, so you do not need to individually test all data source.

The connection is successful if the **HTTP Status** is **200**. If there is an **Error Code** and **Error Message**, the connection failed and further troubleshooting is required.

- d. In the Help sidebar, click **Back to Guided Setup**.
- e. In the Test the connection task section, click **Mark as Complete**.

7. Add multiple instances.

Note: If you do not need to add multiple instances, you can skip this step.

- a. On the left side bar, select the Add Multiple Instances icon (○).
- b. On the Service Graph Connector for SolarWinds page, under the Add Multiple Instances section, select the **Update Data Source Access** task.
- c. On the next page, in the Update Data Source Access section, click **Configure**.
- d. Select the Data Source [sys_data_source] table.
- e. To edit the record, select **Global** from the Scope menu.
- f. Under the **Application Access** tab, select the **Can create**, **Can update**, and **Can delete** check boxes.
- g. Save the record.
- h. From the Scope menu, select **Service Graph Connector for SolarWinds**.
- i. In the Help task bar, click **Mark as Complete**.
- j. Repeat these steps in the Update Scheduled data import access section with the Scheduled data import [scheduled_data_set]

table and the Update Value Access section with the Value [sys_variable_table] table.

8. Clear the cache for the new connection.

- a. Select the **Clear Cache for Datasource and Import set** task, then **Configure**.
- b. Clear the cache by selecting **Global** from the Scope menu.
- c. Enter the following script.

```
GlideTableManager.invalidateTable("sys_data_source");
    GlideCacheManager.flushTable("sys_data_source");

    GlideTableManager.invalidateTable("scheduled_import_set");
    GlideCacheManager.flushTable("scheduled_import_set");

    GlideTableManager.invalidateTable("sys_variable_value");
    GlideCacheManager.flushTable("sys_variable_value");

    GlideTableManager.invalidateTable("sys_db_object");
    GlideCacheManager.flushTable("sys_db_object");
```

- d. Select **Run Script**.
- e. From the Scope menu, select **Service Graph Connector for SolarWinds**.
- f. Click **Mark as Complete**.

9. Add a connection to another SolarWinds instance.

Note: Confirm that the current scope is **Service Graph Connector for SolarWinds**.

- a. In the Add Another Connection section, select **Configure**.

b. Either create or edit a connection.

- To create a new connection, select **Add Connection**.
- To edit an existing connection, select the **Edit** button.

c. On the form, fill in the fields or edit as needed.

Create Connection

Field	Description
Connection Name	Display name for the connection.
Connection URL	Connection Host name for SolarWinds.
User name	Username for SolarWinds authentication.
Password	Password for SolarWinds authentication.

d. Either add or save the connection.

- To add a new connection, select **Create Connection**.
- To save the edits for the existing connection, select **Edit Connection**.

e. Navigate back to the guided setup and click **Mark as Complete**.

f. If needed, set up the MID Server for the connection you created.

- a. Under the Configure Mid Servers section, click **Configure**.
- b. Select the name of the connection you created.
- c. Click the **Use MID server** check box.
- d. Click **Update**.

e. When you're finished with the task, click **Mark as Complete**

g. If needed, configure the connections in the Configure SolarWinds Modules section, by clicking **Configure**.

h. When you're finished, close the window and click **Mark as Complete**.

i. In the Test New Connections section, click **Configure**.

a. Select the name of the data source associated with the newly created connection.

b. Click the **Test Load 20 Records** related link.

Note: If the displayed completion code is Success, then the sources are validated. But if the displayed completion code is Error, then there is an error that you must fix.

c. In the Help sidebar, click **Mark as Complete**.

10. Set up scheduled import jobs.

a. On the left sidebar, click the Set up scheduled import jobs icon (⌚).

b. On the Service Graph Connector for SolarWinds page, in the Set up scheduled import jobs section, select the task **Configure the scheduled import jobs**.

c. In the Configure the scheduled import jobs task section, click **Configure**.

d. Review the fields on the Scheduled Data Import form, which are automatically set.

Scheduled Data Import form

Field	Description
Name	Name of the scheduled job.
Data source	Data source record that defines the data to import.

Field	Description
Run as	Option to run the scheduled job with the credentials of the specified user.
Active	Option to activate the scheduled job. Select this option.
Concurrent Import	Function that loads the data from multiple import sets. The function then processes and transforms the data concurrently.
Partition Method	Partition method for the concurrent import set.
Partition Size	Import set size for early scheduling.
Execute pre-import script	Option to specify a script to run before the import is performed.
Execute post-import script	Option to specify a script to run after the import is performed.
Application	Application containing this record.
Run	Frequency of running the import.
Conditional	Conditions under which this job is executed.

Note: All active SolarWinds scheduled jobs will run in their specified order after the SG-Solarwinds Hardware scheduled job runs. You can modify the **Active** setting for each SolarWinds scheduled job as appropriate for your integration.

- e. Click **Update** if necessary then **Mark as Complete**.

When you complete the guided setup, you can configure the integration to periodically pull data from Solarwinds. The data is saved in tables that extend from the Configuration item [cmdb_ci] table.

The following attributes in the Availability Zone [cmdb_ci_availability_zone] table are populated by collected data:

Attribute label	Attribute name
Name	name
Object ID	object_id

Relationships created for Availability Zone

Parent class	Relationship type	Child class
Availability Zone [cmdb_ci_availability_zone]	Contains::Contained by	Cloud Subnet [cmdb_ci_cloud_subnet]
Availability Zone [cmdb_ci_availability_zone]	Contains::Contained by	Cloud Network [cmdb_ci_network]

The following attributes in the Cloud Key Pair [cmdb_ci_cloud_key_pair] table are populated by collected data:

Attribute label	Attribute name
Object ID	object_id
Name	name

Relationship created for Cloud Key Pair

Parent class	Relationship type	Child class
Cloud Key Pair [cmdb_ci_cloud_key_pair]	Hosted on::Hosts	Logical Datacenter [cmdb_ci_logical_datacenter]

The following attributes in the Cloud Network [cmdb_ci_network] table are populated by collected data:

Attribute label	Attribute name
Name	name
Object ID	object_id

Relationships created for Cloud Network

Parent class	Relationship type	Child class
Cloud Network [cmdb_ci_network]	Contains::Contained by	Cloud Subnet [cmdb_ci_cloud_subnet]
Cloud Network [cmdb_ci_network]	Hosted on::Hosts	Logical Datacenter [cmdb_ci_logical_datacenter]

The following attributes in the Cloud Service Account [cmdb_ci_cloud_service_account] table are populated by collected data:

Attribute label	Attribute name
Account Id	account_id
Name	name
Object ID	object_id
Datacenter Type	datacenter_type

The following attributes in the Cloud Subnet [cmdb_ci_cloud_subnet] table are populated by collected data:

Attribute label	Attribute name
Name	name
Object ID	object_id

The following attributes in the Computer [cmdb_ci_computer] table are populated by collected data:

Attribute label	Attribute name
CPU core count	cpu_core_count
CPU core thread	cpu_core_thread
CPU name	cpu_name
CPU speed (MHz)	cpu_speed
CPU manufacturer	cpu_manufacturer
Is Virtual	virtual

Relationships created for Computer

Parent class	Relationship type	Child class
Computer [cmdb_ci_computer]	Virtualized by::Virtualizes	Virtual Machine Instance [cmdb_ci_vm_instance]
Computer [cmdb_ci_computer]	Contains::Contained by	Disk [cmdb_ci_disk]

The following attributes in the Disk [cmdb_ci_disk] table are populated by collected data:

Attribute label	Attribute name
Model ID	model_id
Device ID	device_id
Name	name
Disk space (GB)	disk_space
Free disk space (GB)	free_space
Size	size
Size bytes	size_bytes
Computer	computer
Manufacturer	manufacturer
Volume serial number	volume_serial_number

Relationship created for Disk

Parent class	Relationship type	Child class
Disk [cmdb_ci_disk]	Reference	Computer [cmdb_ci_computer]

The following attributes in the Hardware [cmdb_ci_hardware] table are populated by collected data:

Attribute label	Attribute name
Model ID	model_id
Name	name
DNS Domain	dns_domain
Manufacturer	manufacturer

Attribute label	Attribute name
Serial number	serial_number
Class	sys_class_name
Default Gateway	default_gateway
Fully qualified domain name	fqdn

Relationships created for Hardware

Parent class	Relationship type	Child class
Hardware [cmdb_ci_hardware]	Owns::Owned by	IP Address [cmdb_ci_ip_address]
Hardware [cmdb_ci_hardware]	Owns::Owned by	Network Adapter [cmdb_ci_network_adapter]

The following attributes in the Hardware Type [cmdb_ci_compute_template] table are populated by collected data:

Attribute label	Attribute name
Name	name
Object ID	object_id

Relationship created for Hardware Type

Parent class	Relationship type	Child class
Hardware Type [cmdb_ci_compute_template]	Hosed on::Hosts	Logical Datacenter [cmdb_ci_logical_datacenter]

The following attributes in the IIS Virtual Directory [cmdb_ci_iisdirectory] table are populated by collected data:

Attribute label	Attribute name
Alias	alias
Installation directory	install_directory
Name	name

Relationship created for IIS Virtual Directory

Parent class	Relationship type	Child class
IIS Virtual Directory [cmdb_ci_iisdirectory]	Runs on::Runs	Hardware [cmdb_ci_hardware]

The following attributes in the Image [cmdb_ci_os_template] table are populated by collected data:

Attribute label	Attribute name
Name	name
Object ID	object_id

Relationship created for Image

Parent class	Relationship type	Child class
Image [cmdb_ci_os_template]	Hosted on::Hosts	Logical Datacenter [cmdb_ci_logical_datacenter]

The following attributes in the IP Address [cmdb_ci_ip_address] table are populated by collected data:

Attribute label	Attribute name
IP Address	ip_address
Netmask	netmask

Attribute label	Attribute name
IP version	ip_version
Nic	nic

Relationship created for IP Address

Parent class	Relationship type	Child class
IP Address [cmdb_ci_ip_address]	Reference	Network Adapter [cmdb_ci_network_adapter]

The following attributes in the Logical Datacenter
[cmdb_ci_logical_datacenter] table are populated by collected data:

Attribute label	Attribute name
Name	name
Region	region
Class	sys_class_name

Relationships created for Logical Datacenter

Parent class	Relationship type	Child class
Logical Datacenter [cmdb_ci_logical_datacenter]	Contains::Contained by	Availability Zone [cmdb_ci_availability_zone]
Logical Datacenter [cmdb_ci_logical_datacenter]	Hosted on::Hosts	Cloud Service Account [cmdb_ci_cloud_service_account]

The following attributes in the Microsoft iis Web Server
[cmdb_ci_microsoft_iis_web_server] table are populated by collected data:

Attribute label	Attribute name
Name	name
Running process command	running_process_command
Version	version
Operational status	operational_status
PID	pid
Install Status	install_status
Type	type

Relationships created for Microsoft iis Web Server

Parent class	Relationship type	Child class
Microsoft iis Web Server [cmdb_ci_microsoft_iis_web_server]	Contains::Contained by	IIS Virtual Directory [cmdb_ci_iisdirectory]
Microsoft iis Web Server [cmdb_ci_microsoft_iis_web_server]	Runs on::Runs	Hardware [cmdb_ci_hardware]

The following attributes in the MS SQL DataBase [cmdb_ci_db_mssql_database] table are populated by collected data:

Attribute label	Attribute name
Data Base	database
Name	name

Relationship created for MS SQL DataBase

Parent class	Relationship type	Child class
MS SQL DataBase [cmdb_ci_db_mssql_database]	Runs on::Runs	Hardware [cmdb_ci_hardware]

The following attributes in the MSFT SQL Instance [cmdb_ci_db_mssql_instance] table are populated by collected data:

Attribute label	Attribute name
Edition	edition
Instance Name	instance_name
Name	name
Operational status	operational_status
Service pack	service_pack
Install Status	install_status

Relationships created for MSFT SQL Instance

Parent class	Relationship type	Child class
MSFT SQL Instance [cmdb_ci_db_mssql_instance]	Runs on::Runs	Hardware [cmdb_ci_hardware]
MSFT SQL Instance [cmdb_ci_db_mssql_instance]	Contains::Contained by	MS SQL DataBase [cmdb_ci_db_mssql_database]

The following attributes in the Network Adapter [cmdb_ci_network_adapter] table are populated by collected data:

Attribute label	Attribute name
DHCP Enabled	dhcp_enabled
Netmask	netmask
Configuration Item	cmdb_ci
Mac manufacturer	mac_manufacturer
MAC Address	mac_address
Name	name

Relationship created for Network Adapter

Parent class	Relationship type	Child class
Network Adapter [cmdb_ci_network_adapter]	Reference	Hardware [cmdb_ci_hardware]

The following attributes in the Serial Number [cmdb_serial_number] table are populated by collected data:

Attribute label	Attribute name
Serial Number	serial_number
Serial Number Type	serial_number_type
Valid	valid

Relationship created for Serial Number

Parent class	Relationship type	Child class
Serial Number [cmdb_serial_number]	Reference	Hardware [cmdb_ci_hardware]

The following attributes in the Software [cmdb_ci_spkg] table are populated by collected data:

Attribute label	Attribute name
Key	key
Name	name
Version	version
Manufacturer	manufacturer

Relationship created for Software

Parent class	Relationship type	Child class
Software [cmdb_ci_spkg]	Reference	Software Instance [cmdb_software_instance]

The following attributes in the Software Instance [cmdb_software_instance] table are populated by collected data:

Attribute label	Attribute name
Install date	install_date
Installed on	installed_on
Name	name

Relationships created for Software Instance

Parent class	Relationship type	Child class
Software Instance [cmdb_software_instance]	Reference	Computer [cmdb_ci_computer]
Software Instance [cmdb_software_instance]	Reference	Hardware [cmdb_ci_hardware]

The following attributes in the Storage Volume [cmdb_ci_storage_volume] table are populated by collected data:

Attribute label	Attribute name
Object ID	object_id
Volume ID	volume_id
Name	name
Size bytes	size_bytes

Relationship created for Storage Volume

Parent class	Relationship type	Child class
Storage Volume [cmdb_ci_storage_volume]	Hosted on::Hosts	Logical Datacenter [cmdb_ci_logical_datacenter]

The following attributes in the Virtual Machine Instance [cmdb_ci_vm_instance] table are populated by collected data:

Attribute label	Attribute name
Name	name
Object ID	object_id
IP Address	ip_address
State	state

Relationships created for Virtual Machine Instance

Parent class	Relationship type	Child class
Virtual Machine Instance [cmdb_ci_vm_instance]	Hosted on::Hosts	Logical Datacenter [cmdb_ci_logical_datacenter]

Parent class	Relationship type	Child class
Virtual Machine Instance [cmdb_ci_vm_instance]	Provisioned From::Provisioned	Image [cmdb_ci_os_template]
Virtual Machine Instance [cmdb_ci_vm_instance]	Provisioned From::Provisioned	Hardware Type [cmdb_ci_compute_template]
Virtual Machine Instance [cmdb_ci_vm_instance]	Use End Point To::Use End Point From	Storage Volume [cmdb_ci_storage_volume]

Service Graph Connector for Tanium (1.5.0)

Use the Service Graph Connector for Tanium to bring in hardware, software, and software usage data from a Tanium environment into your ServiceNow instance.

Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

Supported versions

- Supported versions:
 - Starting from Tanium 1.9 for Hardware and Software
 - Starting from Tanium 1.17 for Software Usage
- Supported ServiceNow versions:
 - San Diego
 - Tokyo

- Utah

Guided Setup

The guided setup for the Service Graph Connector for Tanium provides an organized sequence of tasks to configure the integration on your instance. To access the guided setup, see [Configure guided setup](#).

CMDB Integrations Dashboard

The Integration Commons for CMDB store app provides a dashboard with a central view of the status, processing results, and processing errors of all installed integrations. You can see metrics for all integration runs. You can filter the view to a specific CMDB integration, a specific time duration, or a specific integration run. For more details about monitoring Tanium integrations in the CMDB Integrations Dashboard, see [Using the CMDB Integrations Dashboard](#).

Data Mapping

Data from the Tanium data sources is mapped and transformed into the ServiceNow CMDB Configuration Item (CI) class definitions using the Robust Transform Engine (RTE). Data is inserted into the ServiceNow CMDB using the Identification and Reconciliation Engine (IRE).

You can use the IntegrationHub ETL app to view the data maps. See [IntegrationHub ETL \(3.2\)](#) for more information.

The Tanium data sources include the following:

- SG-Tanium Applications
- SG-Tanium Hardware and Software
- SG-Tanium Usage

Note: SG-Tanium Hardware and Software is supported starting from Tanium 1.9.

When you complete the guided setup, you can configure the integration to periodically pull data from Tanium. The data is loaded into the following staging tables:

- SG Tanium Import [sn_tanium_integ_sg_tanium_import]
- SG Tanium Usage Import [sn_tanium_integ_sg_tanium_usage_import]
- SG-Tanium Applications [sn_tanium_integ_sg_tanium_applications]

The data is then inserted into the following target tables:

- Application [cmdb_ci_appl]
- Computer [cmdb_ci_computer]
- IP Address [cmdb_ci_ip_address]
- Logical Disk [cmdb_ci_file_system]
- Network Adapter [cmdb_ci_network_adapter]
- Physical Disk [cmdb_ci_disk]
- Running Process [cmdb_running_process]
- Software Install [cmdb_sam_sw_install] If SAM is installed.
- Software Instance [cmdb_software_instance] If SAM is not installed.
- Software Package [cmdb_ci_spkg] If SAM is not installed.
- Software Usage [samp_sw_usage]

Note: Software Usage is supported starting from Tanium 1.17.

- TCP [cmdb_tcp]

You need to have the SAM professional plugin (com.snc.samp) installed to have the Software Usage data source appear. You can make the performance faster by adding the following indexes:

- samp_sw_product table prod_name column
- samp_sw_publisher table name column

Note: The indexes are required for instances prior to San Diego. Starting from San Diego, the two indexes are added in the platform code.

Set up scheduled import jobs to pull in data from Tanium into your CMDB.

Before you begin

Dependencies and requirements:

- The [Integration Commons for CMDB](#) store app, which is automatically installed.
- The [CMDB CI Class Models store app](#) store app, which is automatically installed.
- The Datastream Action plugin (com.glide.hub.action_type.datastream), which is automatically installed.

Note: If you have an earlier version of the Service Graph Connector for Tanium, then do not migrate data from the old connector. You must uninstall the previous version and run the new integration.

Starting with the San Diego release, embedded help content will not be visible on the default Polaris theme in the Next Experience. You must activate the embedded help content by completing the following steps:

1. Select a task section in the guided setup, and then click **Configure**.
2. In the menu bar, click the help icon (?

Role required: admin

Procedure

1. Navigate to **All > Service Graph Connector for Tanium > Setup**.
2. On the Getting started page, select **Get Started**.
3. Configure the authentication credentials and HTTP connection.

- a. On the Service Graph Connector for Tanium page, in the Configure the Connection and Credentials section, select the task **Set authentication type**.
 - a. In the **Value** field, enable either token authentication or basic authentication.
 - To enable the token authentication, enter **token**.
 - To enable the basic authentication, enter **basic**.
 - b. Click **Update**, then **Mark as Complete**.
- b. In the Configure the Token auth/Basic auth Credentials section, configure your credentials.
 - a. Select **Configure**.
 - b. In the **Name** field, enter a name for the authentication.

For example, **Tanium credentials**.

 - For a Token auth, in the **API Key** field, enter your Tanium token.
 - For a Basic auth, in the **User name** and **Password** field, enter your Tanium user name and password.
 - c. Click **Update**, then **Mark as Complete**.
- c. In the Configure the Connection section, configure the connections.
 - a. Click **Configure**.
 - b. Update the **Host** field with a Tanium base URL or IP address.

For example, **demojamfhost.com** or **127.0.0.1**. If you are using a Tanium cloud instance, enter API at the end of your base URL, such as **demojamfhost-API.com**.
 - c. If you are using anything other than **https**, then update the **Protocol** field.

Note: If the Tanium server connection requires a MID Server setup, select the **Use MID Server** check box and select the MID Server. For more information about the Tanium API, see the [Tanium Developer documentation](#).

- d. Click **Update**, then **Mark as Complete**.
- d. In the Configure the View section, to create a custom view.

Note: The integration depends on a custom view from Tanium.

 - a. Click **Configure**.
 - b. In Tanium, navigate to **Modules > Asset**.
 - c. In the left navigation menu, select **Views**.
 - d. If you have not done so already, create the ServiceNow (reserved) view by selecting **Create View > Create ServiceNow view**.

Note: By default, the ServiceNow (reserved) view is non-editable.

 - e. Create a copy of the view to edit it.
 - f. Edit the copy.
 - a. Add all fields from the **SIU Product Usage** bucket.
 - b. Add **Asset** to the **Last Seen** field.
 - c. Add **Network Adapter** to the **Model** field.
 - g. Save the view.
 - h. In the ServiceNow instance, select the view you created in the drop-down menu.
 - i. Click the **Set View** button.
 - j. Click **Mark as Complete**.

- e. In the ADM setup instruction section, ignore the instructions as application dependency mapping is not supported currently.
- f. In the Test the Connection section, test the connection by selecting **Configure**.
When the test is finished, select **Mark as Complete**.

Testing the connection may take a few moments. The page is refreshed to show the test results.

Note: The connection is successful if the **HTTP Status** is **200**. If there is anything displayed in the **Error Code** and **Error Message** fields, then the connection failed and further troubleshooting is required.

4. Configure the Tanium scheduled job.

- a. In the Configure the Scheduled Import section, click **Configure**.
- b. Select the scheduled data import that you want to activate.
- c. On the form, review the fields as needed.

Scheduled Data Import form

Field	Description
Name	Name of the scheduled job.
Data source	Data source record that defines the data to import.
Run as	Option to run the scheduled job with the credentials of the specified user.
Active	Option to activate the scheduled job. Select this option.
Concurrent Import	Function that loads the data from multiple import sets. The function then processes

Field	Description
	and transforms the data concurrently.
Partition Method	Partition method for the concurrent import set.
Partition Size	Import set size for early scheduling.
Execute pre-import script	Option to specify a script to run before the import is performed.
Execute post-import script	Option to specify a script to run after the import is performed.
Application	Application that contains this scheduled job.
Run	Frequency of running the import. Set this value to how frequent you want to pull your data.
Conditional	Conditions under which this job is executed.

- d. Click **Execute Now**.
 - e. In the Configure the Scheduled Import task section, click **Mark as Complete**.
5. Add multiple instances.
- Note:** If you do not need to add multiple instances, you can skip this step.
- a. On the left side bar, select the Add Multiple Instances icon ().

- b. On the Service Graph Connector for Tanium page, under the Add Multiple Icons section, select the **Update Data Source Access** task.
 - c. On the next page, in the Update Data Source Access section, click **Configure**.
 - d. Select the Data Source [sys_data_source] table.
 - e. Edit the record by selecting **Global** from the Scope menu.
 - f. Under the Application Access tab, select the **Can create**, **Can update**, and **Can delete** check boxes.
 - g. Save the record.
 - h. From the Scope menu, select **Service Graph Connector for Tanium**.
 - i. In the Help task bar, click **Mark as Complete**.
 - j. Repeat these steps in the Update Scheduled data import access section with the Scheduled data import [scheduled_data_set] table.
6. Clear the cache for the new connection.
 - a. Select the **Clear Cache for Datasource and Import set** task, then **Configure**.
 - b. Clear the cache by selecting **Global** from the Scope menu.
 - c. Enter the following script.

```
GlideTableManager.invalidateTable("sys_data_source");
);
    GlideCacheManager.flushTable("sys_data_source");

    GlideTableManager.invalidateTable("scheduled_import_set");
    GlideCacheManager.flushTable("scheduled_import_set");

    GlideTableManager.invalidateTable("sys_db_obj")
```

```
ect");
GlideCacheManager.flushTable("sys_db_object");
```

- d. Select **Run Script**.
 - e. From the Scope menu, select **Service Graph Connector for Tanium**.
 - f. Click **Mark as Complete**.
7. Add a basic auth connection to another Tanium instance by selecting the **Add Another Basic Auth Connection** task.

Note: Confirm that the current scope is **Service Graph Connector for Tanium**.

- a. In the Add Another Connection section, select **Configure**.
- b. Either create or edit a connection.
 - To create a new connection, select **Add Connection**.
 - To edit an existing connection, select the **Edit** button.
- c. On the form, fill in the fields or edit as needed.

Create Connection form

Field	Description
Connection Name	Display name for the connection.
Connection URL	Connection Host name for Tanium.
User name	Username for Tanium authentication.
Password	Password for Tanium authentication.

- d. Either add or save the connection.
 - To add a new connection, select **Create Connection**.

- To save the edits for the existing connection, select **Edit Connection**.
- e. Navigate back to the guided setup and click **Mark as Complete**.
- f. Repeat these steps in the Add Another Token Auth Connection section with the following information.

Create Connection form

Field	Description
Connection Name	Display name for the connection.
Host Name	Host name for Tanium.
Token	Token name for Tanium.

- g. If needed, set up the MID Server for the connection you created.
 - a. Under the Configure Mid Servers section, click **Configure**.
 - b. Select the name of the connection you created.
 - c. Click the **Use MID server** check box.
 - d. Click **Update**.
 - e. When you're finished with the task, click **Mark as Complete**
- 8. Configure the sets of the data sources and scheduled data imports for the new connection.
 - a. In the Configure Data Sources and Scheduled Imports section, click **Configure**.
 - b. On the form, enter the following values.

Field	Value
Connection and Credential Alias	Select the connection alias that was created in the previous step.

Field	Value
View	Select the Tanium view you want to load.

- c. Click **Generate Data source and Scheduled import**, then click **Mark as Complete**.
9. Configure the Tanium scheduled job.
 - a. In the Configure the Scheduled Imports section, click **Configure**.
 - b. Select the scheduled data import that you want to activate.
 - c. On the form, review the fields as needed.

Scheduled Data Import form

Field	Description
Name	Name of the scheduled job.
Data source	Data source record that defines the data to import.
Run as	Option to run the scheduled job with the credentials of the specified user.
Active	Option to activate the scheduled job. Select this option.
Concurrent Import	Function that loads the data from multiple import sets. The function then processes and transforms the data concurrently.
Partition Method	Partition method for the concurrent import set.

Field	Description
Partition Size	Import set size for early scheduling.
Execute pre-import script	Option to specify a script to run before the import is performed.
Execute post-import script	Option to specify a script to run after the import is performed.
Application	Application that contains this scheduled job.
Run	Frequency of running the import. Set this value to how frequent you want to pull your data.
Conditional	Conditions under which this job is executed.

- d. Click **Execute Now**.
 - e. Repeat these substeps for each of the scheduled data imports.
 - f. In the Configure the Scheduled Import task section, click **Mark as Complete**.
10. On the left side bar, click the advanced icon (○) then in the Advanced section, select the **Advanced Settings** task.
 11. Set up your advanced settings.
 - a. In the Advanced Settings section, click **Configure**.
 - b. If you do not want to exclude the serial number population, enter **false** in the **Exclude the serial number population** field.
 - c. Click **Save**.

- d. Close the window and click **Mark as Complete**.

When you complete the guided setup, you can configure the integration to periodically pull data from Tanium. The data is saved in tables that extend from the Configuration item [cmdb_ci] table.

The following attributes in the Computer [cmdb_ci_computer] table are populated by collected data:

Attribute label	Attribute name
Name	name
Serial number	serial_number
Class	sys_class_name
CPU core count	cpu_core_count
CPU count	cpu_count
CPU speed (MHz)	cpu_speed
CPU type	cpu_type
DNS Domain	dns_domain
IP Address	ip_address
CPU manufacturer	cpu_manufacturer
CPU name	cpu_name
Is Virtual	virtual
Most recent discovery	last_discovered
Operating System	os
OS Domain	os_domain
OS Service Pack	os_service_pack

Attribute label	Attribute name
OS Version	os_version
RAM (MB)	ram
Model ID	model_id
Manufacturer	manufacturer

Relationships created for Computer

Parent class	Relationship type	Child class
Computer [cmdb_ci_computer]	Owns::Owned by	Network Adapter [cmdb_ci_network_adapter]
Computer [cmdb_ci_computer]	Owns::Owned by	IP Address [cmdb_ci_ip_address]
Computer [cmdb_ci_computer]	Contains::Contained by	Disk [cmdb_ci_disk]
Computer [cmdb_ci_computer]	Contains::Contained by	File System [cmdb_ci_file_system]

The following attributes in the Disk [cmdb_ci_disk] table are populated by collected data:

Attribute label	Attribute name
Manufacturer	manufacturer
Name	name
Computer	computer
Serial number	serial_number
Device interface	device_interface

Attribute label	Attribute name
Storage type	storage_type
Model ID	model_id
Device ID	device_id
Size bytes	size_bytes

Relationship created for Disk

Parent class	Relationship type	Child class
Disk [cmdb_ci_disk]	Reference	Computer [cmdb_ci_computer]

The following attributes in the File System [cmdb_ci_file_system] table are populated by collected data:

Attribute label	Attribute name
Name	name
File system	file_system
Free space bytes	free_space_bytes
Label	label
Media type	media_type
Mount point	mount_point
Size bytes	size_bytes
Computer	computer

Relationship created for File System

Parent class	Relationship type	Child class
File System [cmdb_ci_file_system]	Reference	Computer [cmdb_ci_computer]

The following attributes in the IP Address [cmdb_ci_ip_address] table are populated by collected data:

Attribute label	Attribute name
Nic	nic
IP Address	ip_address
IP version	ip_version
Name	name

Relationship created for IP Address

Parent class	Relationship type	Child class
IP Address [cmdb_ci_ip_address]	Reference	Network Adapter [cmdb_ci_network_adapter]

The following attributes in the Network Adapter [cmdb_ci_network_adapter] table are populated by collected data:

Attribute label	Attribute name
Discovery source	discovery_source
Netmask	netmask
Mac manufacturer	mac_manufacturer
MAC Address	mac_address
Name	name

Attribute label	Attribute name
Model ID	model_id
DHCP Enabled	dhcp_enabled

Relationship created for Network Adapter

Parent class	Relationship type	Child class
Network Adapter [cmdb_ci_network_adapter]	Reference	Computer [cmdb_ci_computer]

The following attributes in the Serial Number [cmdb_serial_number] table are populated by collected data:

Attribute label	Attribute name
Serial Number	serial_number
Serial Number Type	serial_number_type
Valid	valid

The following attributes in the Software [cmdb_ci_spkg] table are populated by collected data:

Attribute label	Attribute name
Version	version
Manufacturer	manufacturer
Key	key
Name	name

Relationship created for Software

Parent class	Relationship type	Child class
Software [cmdb_ci_spkg]	Reference	Computer [cmdb_ci_computer]

The following attributes in the Software Instance [cmdb_software_instance] table are populated by collected data:

Attribute label	Attribute name
Name	name
Installed on	installed_on

Relationship created for Software Instance

Parent class	Relationship type	Child class
Software Instance [cmdb_software_instance]	Reference	Computer [cmdb_ci_computer]

The following attributes in the Application [cmdb_ci_appl] table are populated by collected data:

Attribute label	Attribute name
Class	sys_class_name
Name	name
Running process command	running_process_command

Relationship created for Application

Parent class	Relationship type	Child class
Application [cmdb_ci_appl]	Runs on::Runs	Computer [cmdb_ci_computer]

Service Graph Connector for VMware Workspace ONE UEM (1.6.0)

Use the Service Graph Connector for VMware Workspace ONE UEM to pull data from VMware Workspace ONE Unified Endpoint Management (UEM) into your ServiceNow Instance.

Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

The integration imports different hardware assets into the ServiceNow® Configuration Management Database (CMDB) application.

Supported versions

- Supported versions:
 - VMware Workspace ONE UEM version 2008
 - Application Discovery Manager (ADM) API version 2
- Supported ServiceNow versions:
 - Quebec
 - Rome
 - San Diego
 - Tokyo

Use Cases

The following are examples on how you can use the Service Graph Connector for different ServiceNow applications:

- [IT Operations Management \(ITOM\) Visibility](#)

- Detailed hardware and application inventory for Android, Apple, and Windows mobile devices. The inventory can be used with or without Software Asset Management (SAM).
- Compliance tracking for mobile devices. You can build your own device (BYOD) or use corporate-owned devices.
- [IT Service Management \(ITSM\)](#)
 - Incident, problem, change on discovered configuration items (CI).
 - Ownership tracking and assignment for mobile devices.

Guided Setup

The guided setup for the Service Graph Connector for VMware Workspace ONE UEM provides an organized sequence of tasks to configure the integration on your instance. To access the guided setup, see [Configure guided setup](#).

CMDB Integrations Dashboard

The Integration Commons for CMDB store app provides a dashboard with a central view of the status, processing results, and processing errors of all installed integrations. You can see metrics for all integration runs. You can filter the view to a specific CMDB integration, a specific time duration, or a specific integration run. For more details about monitoring Workspace ONE integrations in the CMDB Integrations Dashboard, see [Using the CMDB Integrations Dashboard](#).

Data Mapping

Data from the VMware Workspace ONE UEM Devices and Apps data sources is mapped and transformed into the ServiceNow CMDB Configuration Item (CI) class definitions using the Robust Transform Engine (RTE). Data is inserted into the ServiceNow CMDB using the Identification and Reconciliation Engine (IRE).

When you complete the guided setup, you configure the integration to periodically pull data from VMware Workspace ONE UEM.

You can use the IntegrationHub ETL app to view the data maps. See [IntegrationHub ETL \(3.2\)](#) for more information.

The data is loaded into the SG-Workspace ONE UEM Devices and Apps staging [sn_vmwoneuem_integ_devices_and_apps] table.

The data is then inserted into the following target tables:

- Computer [cmdb_ci_computer]
- Handheld Computing Device [cmdb_ci_handheld_computing]
- Media Player [cmdb_ci_media_player]
- Network Adapter [cmdb_ci_network_adapter]
- Printer [cmdb_ci_printer]
- SAM Software Installation [cmdb_sam_sw_install], if com.snc.sams plugin is installed.
- Serial Number [cmdb_serial_number]
- Software Instance [cmdb_software_instance], if com.snc.sams plugin is not installed.
- Software Package [cmdb_ci_spkg], if com.snc.sams plugin is not installed.

Note: To view any additional information such as the device owner, type of ownership, or compliance status, you need to switch to the SG-Workspace ONE UEM view. This view will display a **SG-Workspace ONE UEM Device Related** tab in the related list tabs with the additional information.

Use the Service Graph Connector for VMware Workspace ONE UEM to pull mobile and computing devices data from VMware Workspace ONE Unified Endpoint Management (UEM) into your ServiceNow instance.

Before you begin

To use this Service Graph Connector, you need a subscription to a Subscription Unit that is based in the IT Operations Management (ITOM) Visibility application or in the ITOM Discovery application. As defined in the section titled "Managed IT Resource Types" in [ServiceNow Subscription Unit Overview](#), for managed IT resources that are created or modified in the CMDB by this Service Graph Connector, but that are

not yet managed by [ITOM Visibility](#) or [ITOM Discovery](#), these resources will increase Subscription Unit consumption from that application. Review your current Subscription Unit consumption within ITOM Visibility or ITOM Discovery to ensure available capacity.

Dependencies and requirements:

- The [Integration Commons for CMDB](#) store app, which is automatically installed.
- The [CMDB CI Class Models](#) store app store app, which is automatically installed.
- The ITOM Discovery License plugin (com.snc.item.discovery.license). You must activate this plugin.
- ITOM Licensing plugin (com.snc.item.license). For more information, see [Request Discovery](#).
- The Datastream Action plugin (com.glide.hub.action_type.datastream), which is automatically installed.

Starting with the San Diego release, embedded help content will not be visible on the default Polaris theme in the Next Experience. You must activate the embedded help content by completing the following steps:

1. Select a task section in the guided setup, and then click **Configure**.
2. In the menu bar, click the help icon (?

Roles required: admin

About this task

To configure the Service Graph Connector for VMware Workspace ONE UEM, you must configure your OAuth authentication credentials (step 4) if you have these credentials. If you don't have these credentials, then you must configure your Basic authentication credentials (step 5). Do not configure both OAuth and Basic credentials.

Procedure

1. Navigate to **All > Service Graph Connector VMware Workspace > Setup**.
2. On the Getting started page, select **Get Started**.
3. Configure your OAuth authentication credentials.
If you do not have OAuth credentials, skip this step and configure the Basic authentication credentials in step 5.
 - a. On the Service Graph Connector for VMware Workspace ONE UEM page, in the Configure the connection section, select the task **Configure authentication credentials**.
 - b. On the next page, in the Configure authentication credentials section, select **Configure**.
 - c. On the form, fill in the following fields.

Application Registries form

Field	Description
Client ID	Client ID of the VMware Workspace ONE UEM console.
Client Secret	Client secret of the VMware Workspace ONE UEM console. Note: You can click the lock icon () to view the client secret.
Token URL	The Token URL of VMware Workspace ONE UEM console so that you can fetch the access token.

Field	Description
	<p>Note: For more information about the Token URL, see the VMware knowledge base article on the VMware documentation site.</p>

To get more information about how to get OAuth credentials, see the [VMware documentation site](#).

- d. Review the other fields on the Applications Registries form as needed.

Application Registries form

Field	Description
Name	Name of the OAuth app.
OAuth API Script	Script that is used to customize requests and responses to the external OAuth provider.
Logo URL	Logo URL for the OAuth app.
Default Grant type	The Default Grant Type that is used to establish the OAuth token.
Refresh Token Lifespan	Number of seconds that a refresh token issued will be good for.
PKCE required	Option to enable public clients to require PKCE during the authorization flow.
Application	Application that contains this record.

Field	Description
Accessible from	Location where the OAuth is accessible from.
Active	Option to activate the OAuth app.
Authorization URL	OAuth authorization code end-point.
Token Revocation URL	OAuth access token revocation end-point.
Redirect URL	The OAuth app end-point to receive authorization code.
Use mutual authentication	Option to use mutual authentication for token requests and revocations. This option requires that a Mutual Auth Profile is specified.
Send Credentials	Option to enable the OAuth Client to populate client credentials in the request.
Comments	Comments about the OAuth app.

- e. Click **Update** if necessary.
- f. In the Configure authentication credentials task section, click **Mark as Complete**.
4. Configure your Basic authentication credentials.
If the OAuth credentials were configured in step 4, skip this step.
- a. On the left side bar, click the Configure the Basic Auth connection icon () and select the task **Select authentication type**.

- b. On the next page, in the Set authentication type section, click **Configure**.
 - c. Update the **Value** field to `basic`.
 - d. In the Set authentication type section, select **Mark as Complete**.
 - e. In the Configure authentication credentials section, click **Configure** and do the following:
 - a. In the **Name** field, enter a name for the authentication. For example, `VMware Workspace ONE UEM Basic credentials`.
 - b. In the **User name** field, enter your VMware Workspace ONE UEM user name.
 - c. In the **Password** field, enter your VMware Workspace ONE UEM password.
 - d. Click **Update**.
 - f. In the Configure API key section, click **Configure**, in the **API Key** field, enter your VMware Workspace ONE UEM tenant code, and then click **Update**.
5. Configure the HTTP connection.
 - a. In the Configure HTTP connection task section, click **Configure**.
 - b. On the form, fill in the fields.

HTTP(s) Connection form

Field	Description
Name	Name of the connection.
Use MID server	MID Server that sends this HTTP connection. Using a MID Server is not compatible with mutual authentication.
Host	Target host value that is used by the connection. The Connection URL will

Field	Description
	<p>automatically fill in the hostname.</p> <p>Note: Update the Host field with a VMware Workspace ONE UEM base URL. For example, <code>as4855.awmdm.com</code>.</p>
Credential	Credential value used by this connection.
Connection alias	Connection value that is used to refer to the connection.
URL builder	URL builder that is used to build the connection URL.
Connection URL	Connection URL of the connection. You can either manually enter your connection URL or use the URL builder to build the connection string.
Mutual authentication	Optional to enable mutual authentication.
Protocol	Underlying protocol used by the connection. <p>Note: Update the Protocol field if you are using anything other than <code>https</code>.</p>
Active	Option to activate the HTTP connection.

Field	Description
Domain	Domain that contains the connection.
Override default port	Target value port that is used by the connection.
Base path	Base path for HTTP(s) connection. Note: You do not need to update this field.

Note: The HTTP connection will be pre-configured to use the authentication credentials that were configured during the previous setup task.

- c. Click **Update** if necessary.
 - d. In the Configure HTTP Connection task section, click **Mark as Complete**.
6. Validate the data sources.
- a. In the Validate data sources task section, click **Configure**.
 - b. Review the fields on the Data Source form, which is automatically set.

Data Source form

Field	Description
Name	Unique name of this data source.
Import set table label	Specify the import set table that is produced by this data source.
Import set table name	Name of the table that will be created for this data source.

Field	Description
Type	Data storage type of the data to be imported.
Data in single column	Data in single column.
Use Batch Import	Option to use batch insert to the import set table.
Application	Application that contains this record.
Data Stream action	Data Stream action that provides complex object streams to load data.
Data Loader	Script that loads data in the import set table.

- c. Test the connection by clicking the **Test Load 20 Records** related link.

Testing the connection takes a few moments, after which the page refreshes to show the test results. The connection is successful if the **HTTP Status** is **200**. If there is an **Error Code** and **Error Message**, then the connection failed and further troubleshooting is required.

Note: Do not click **Load All Records** during this setup.

- d. In the Help sidebar, click **Back to Guided Setup**.

- e. In the Validate data sources task section, click **Mark as Complete**.

7. (Optional) Configure additional settings.

- a. On the left side bar, click the Configure additional settings icon ().

- b. On the Service Graph Connector for VMware Workspace ONE UEM page, in the Configure additional settings section, select the task **Configure duplicate detection rules**.

- c. In the Configure duplicate detection rules section, click **Configure**.
- d. On the CMDB Duplicate Row Rules form, update the Active column value to **true** to activate the duplicate detection rule.
Note: To remove fields from being evaluated, add the field names with a comma in a separated list in the Ignore Fields column.
- e. In the Help side bar, select **Mark as Complete**.
- f. In the Set software import section, click **Configure**.
- g. In the **Value** field, enter **false** to import the software data then close the window.
- h. In the Import non-managed software section, click **Configure**.
- i. In the **Value** field, enter **false** to include non-managed software then close the window.
- j. In the Import apps with status section, click **Configure**.
- k. Add the status of the applications you want to import by updating the **Value** field.
By default, the connector imports applications labeled as Installed, Pending Removal, and Unknown.

Status values of applications

Status	Value
Pending Install	1
Installed	2
Pending Removal	3
Removed	4
Unknown	5

- I. Close the window and click **Mark as Complete**.

8. Set up the scheduled import jobs.

- a. On the left side bar, click the **Set up scheduled import jobs** button.
- b. On the Service Graph Connector for VMware Workspace ONE UEM page, in the Set up scheduled import jobs section, select the task **Configure the scheduled job**.
- c. In the Configure the scheduled job task section, click **Configure**.
- d. Select the name of the scheduled import that you want to run.
- e. Review the pre-populated fields on the Scheduled Data Import form.

Scheduled Data Import form

Field	Description
Name	Name of the scheduled job.
Data source	Data source record that defines the data to import.
Run as	Option to run the scheduled job with the credentials of the specified user.
Active	Option to activate the scheduled job. Select this option.
Concurrent Import	Function that loads the data from multiple import sets. The function then processes and transforms the data concurrently.
Partition Method	Partition method for the concurrent import set.
Partition Size	Import set size for early scheduling.

Field	Description
Execute pre-import script	Option to specify a script to run before the import is performed.
Execute post-import script	Option to specify a script to run after the import is performed.
Application	Application that contains this scheduled job.
Run	Frequency of running the import.
Conditional	Conditions under which this job is executed.

- f. Click **Execute Now** and repeat the **Configure the scheduled job** task and these substeps for the other imports if needed.
- g. In the Help task bar, click **Mark as Complete**.

If you need to connect to multiple instances of VMware Workspace ONE Unified Endpoint Management (UEM), then create multiple connections and scheduled imports to import data from multiple data servers.

Before you begin

Ensure that you are in the Service Graph Connector for VMware Workspace ONE UEM application scope.

Role required: admin

Procedure

1. Navigate to **All > Service Graph Connector VMWare Workspace > Setup**.
2. On the left side bar, click the **Create another OAuth connection** icon .

3. On the Service Graph Connector for VMware Workspace ONE UEM page, in the Create another connection section, select the task **Update Scheduled Data Import access** and do the following:
 - a. In the Update Scheduled Data Import access task section, select **Configure**.
 - b. Select the Scheduled Data Import [scheduled_import_set] table.
 - c. To edit the record, select **Global** from the Scope menu.
 - d. Under the **Application Access** tab, select the **Can create**, **Can update**, and **Can delete** check boxes.
 - e. Save the record.
 - f. From the Scope menu, select **Service Graph Connector for VMWare Workspace ONE UEM**.
 - g. In the Help task bar, click **Mark as Complete**.
4. On the Service Graph Connector for VMware Workspace ONE UEM page, in the Create another connection section, select the task **Update Data Source Access** and do the following:
 - a. In the Update Scheduled Data Import access task section, select **Configure**.
 - b. Select the Data Source [sys_data_source] table.
 - c. To edit the record, select **Global** from the Scope menu.
 - d. Under the **Application Access** tab, select the **Can create**, **Can update**, and **Can delete** check boxes.
 - e. Save the record.
 - f. From the Scope menu, select **Service Graph Connector for VMWare Workspace ONE UEM**.
 - g. In the Help task bar, click **Mark as Complete**.
5. On the Service Graph Connector for VMware Workspace ONE UEM page, in the Create another OAuth connection section, select the task **Clear Cache for Datasource and Import set** and do the following:

- a. To clear the cache, select **Global** from the Scope menu.
- b. In the Clear Cache for Datasource and Import Set, select **Configure**.
- c. Enter the following script.

```
GlideTableManager.invalidateTable("sys_data_source");
    GlideCacheManager.flushTable("sys_data_source");

    GlideTableManager.invalidateTable("scheduled_import_set");
    GlideCacheManager.flushTable("scheduled_import_set");

    GlideTableManager.invalidateTable("sys_db_object");
    GlideCacheManager.flushTable("sys_db_object");
```

- d. Select **Run Script**.
 - e. From the Scope menu, select **Service Graph Connector for VMWare Workspace ONE UEM**.
 - f. Click **Mark as Complete**.
6. On the Service Graph Connector for VMware Workspace ONE UEM page, in the Create another OAuth connection section, do the following:

Note: You need to the following information from your VMware Workspace ONE UEM administrator:

- Client ID
- Client Secret
- Token URL
- Connection URL

To get more information about how to get OAuth credentials, see the [VMware documentation](#) on the VMware documentation site. To get more information about the Token URL, see the [Workspace ONE Access Token URL documentation](#) on the VMware documentation site.

- a. In the Create or Edit connection, select **Configure**.
- b. Do one of the following:
 - To create a new connection, select **Add Connection**.
 - To edit an existing connection, select the **Edit** button.
- c. On the form, fill in the fields or edit as needed.

Connection form

Field	Description
Connection Name	Display name for the connection.
Connection URL	Connection URL for the new connection.
Use MID Server	Option to select a MID Server that sends this connection. Using a MID Server is not compatible with mutual authentication.

Field	Description
MID Server	MID Server for the connection.
OAuth Entity Name	Display name for OAuth Entity.
OAuth Client ID	Client ID for the provider.
OAuth Client Secret	Client Secret for the provider.
OAuth Token URL	Callback URL for the provider.

- d. Do one of the following:
- To create the new connection, select **Create and Get OAuth Tokens**.
 - To save the edits for the existing connection, select **Edit and Get OAuth Token**.
- e. When you're finished, select **Mark as Complete**.
7. On the Service Graph Connector for VMware Workspace ONE UEM page, in the Create another OAuth connection section, select the task **Create data sources and scheduled data imports** and do the following:
- a. In the Create data sources and scheduled data imports section, select **Configure**.
 - b. On the form, fill in the fields.

SG-Workspace ONE UEM Create Data Source and Scheduled Import form

Field	Description
Scheduled Data source and Import name prefix	Contents of this field that will be prepended to the Scheduled Data Import name.
Connection and Credential Alias	Connection and Credential Alias of the import. Select

Field	Description
	the connection alias that was created in the previous step.
Run Scheduled Imports as User	User who will run the scheduled data import.

- c. In the Help task bar, click **Mark as Complete**.

When you complete the guided setup, you can configure the integration to periodically pull data from VMware Workspace ONE Unified Endpoint Management (UEM). The data is saved in tables that extend from the Configuration item [cmdb_ci] table.

The following attributes in the Computer [cmdb_ci_computer] table are populated by collected data:

Attribute label	Attribute name
Name	name
Serial number	serial_number
Is Virtual	virtual
Most recent discovery	last_discovered
Operating System	os
OS Version	os_version
RAM (MB)	ram
Assigned to	assigned_to
Model ID	model_id
Manufacturer	manufacturer

The following attributes in the Handheld Computing Device [cmdb_ci_handheld_computing] table are populated by collected data:

Attribute label	Attribute name
MAC Address	mac_address
Name	name
Serial number	serial_number
IMEI	imei
Most recent discovery	last_discovered
Operating System	os
OS Version	os_version
Phone Number	phone_number
RAM (MB)	ram
Root Access	root_access
Manufacturer	manufacturer
Model ID	model_id
Carrier	carrier
Assigned to	assigned_to

The following attribute in the Hardware [cmdb_ci_hardware] table is populated by collected data:

Attribute label	Attribute name
Most recent discovery	last_discovered

Relationships created for Hardware

Parent class	Relationship type	Child class
Hardware [cmdb_ci_hardware]	Owns::Owned by	Network Adapter [cmdb_ci_network_adapter]
Hardware [cmdb_ci_hardware]	Owns::Owned by	IP Address [cmdb_ci_ip_address]
Hardware [cmdb_ci_hardware]	Reference	SG-Workspace ONE UEM Device Related [sn_vmwoneuem_integ_device_related]
Hardware [cmdb_ci_hardware]	Reference	Key Value [cmdb_key_value]

The following attributes in the IP Address [cmdb_ci_ip_address] table are populated by collected data:

Attribute label	Attribute name
Name	name
Nic	nic
IP Address	ip_address
IP version	ip_version

Relationship created for IP Address

Parent class	Relationship type	Child class
IP Address [cmdb_ci_ip_address]	Reference	Network Adapter [cmdb_ci_network_adapter]

The following attributes in the Key Value [cmdb_key_value] table are populated by collected data:

Attribute label	Attribute name
Key	key
Value	value

The following attributes in the Media Player [cmdb_ci_media_player] table are populated by collected data:

Attribute label	Attribute name
Name	name
Serial number	serial_number
Manufacturer	manufacturer
Model ID	model_id
MAC Address	mac_address
Assigned to	assigned_to
Most recent discovery	last_discovered

The following attributes in the Network Adapter [cmdb_ci_network_adapter] table are populated by collected data:

Attribute label	Attribute name
Name	name
Configuration Item	cmdb_ci
MAC Address	mac_address
Most recent discovery	last_discovered

Relationship created for Network Adapter

Parent class	Relationship type	Child class
Network Adapter [cmdb_ci_network_adapter]	Reference	Hardware [cmdb_ci_hardware]

The following attributes in the Printer [cmdb_ci_printer] table are populated by collected data:

Attribute label	Attribute name
Assigned to	assigned_to
MAC Address	mac_address
Name	name
Most recent discovery	last_discovered
Manufacturer	manufacturer
Model ID	model_id
Serial number	serial number

The following attributes in the Serial Number [cmdb_serial_number] table are populated by collected data:

Attribute label	Attribute name
Serial Number	serial_number
Serial Number Type	serial_number_type
Valid	valid

Relationships created for Serial Number

Parent class	Relationship type	Child class
Serial Number [cmdb_serial_number]	Reference	Computer [cmdb_ci_computer]
Serial Number [cmdb_serial_number]	Reference	Handheld Computing Device [cmdb_ci_handheld_c omputing]
Serial Number [cmdb_serial_number]	Reference	Printer [cmdb_ci_printer]
Serial Number [cmdb_serial_number]	Reference	Media Player [cmdb_ci_media_play er]

The following attributes in the SG-Workspace ONE UEM Device Related [sn_vmwoneuem_integ_device_related] table are populated by collected data:

Attribute label	Attribute name
Device ID	device_id
Device Compliance State	compliance_state
Device Ownership	device_ownership
User Email	user_email
User Name	user_name

The following attributes in the Software [cmdb_ci_spkg] table are populated by collected data:

Attribute label	Attribute name
Key	key

Attribute label	Attribute name
Name	name
Version	version

Relationship created for Software

Parent class	Relationship type	Child class
Software [cmdb_ci_spkg]	Reference	Software Instance [cmdb_software_instance]

The following attributes in the Software Instance [cmdb_software_instance] table are populated by collected data:

Attribute label	Attribute name
Name	name
Installed on	installed_on

Relationship created for Software Instance

Parent class	Relationship type	Child class
Software Instance [cmdb_software_instance]	Reference	Hardware [cmdb_ci_hardware]

CMDB 360/Multisource CMDB

CMDB 360 retains complete history about discovery sources and proposed values, involved in updates of CI attributes. Use CMDB 360 data to track how the CMDB is populated by various discovery sources at the CI attribute level. Also, to revert CI updates from a specific discovery source, or to recompute attribute values using updated reconciliation rules.

Starting with the Utah release, the Multisource CMDB feature is part of the CMDB 360 feature. CMDB 360 provides all the functionality of Multisource

CMDB and additional capabilities such as an analytics dashboard, and new query functionality. You can access all of the CMDB 360 capabilities in the [CMDB 360 view in CMDB Workspace](#).

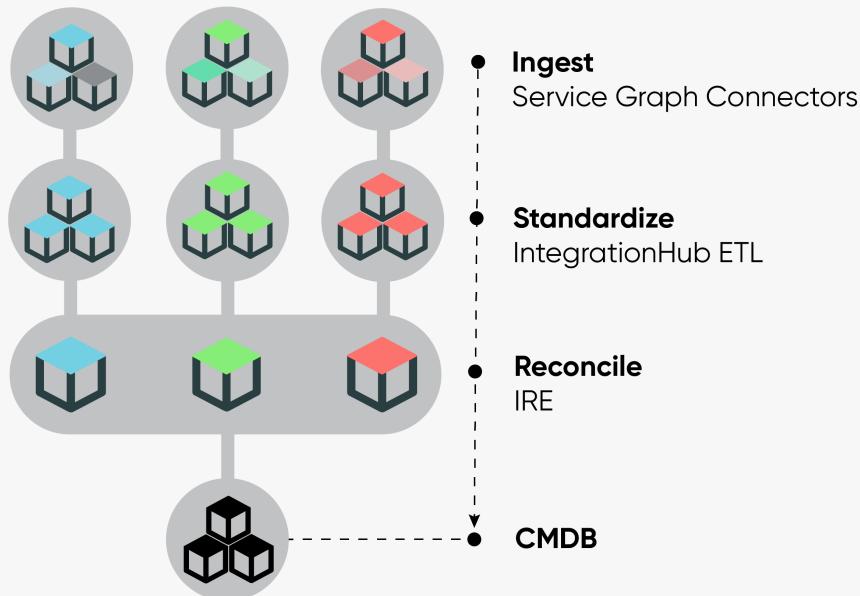
How CMDB 360 works

When multiple discovery sources attempt to update the same CI attribute, the [Identification and Reconciliation Engine \(IRE\)](#) uses reconciliation rules to select a single discovery source for the update. Without CMDB 360, details about the lower-priority discovery sources whose values were rejected, are discarded. Also, it is difficult to identify the source of an attribute value without CMDB 360.

With CMDB 360, the raw details for every discovery source and CI combination are retained for both, discovery sources that were selected for an update and all others that were not. CMDB 360 data, consisting of records for each discovery source and CI combination, is stored in the CMDB MultiSource Data [cmdb_multisource_data] table. You can examine, query, and report on the CMDB 360 data store.

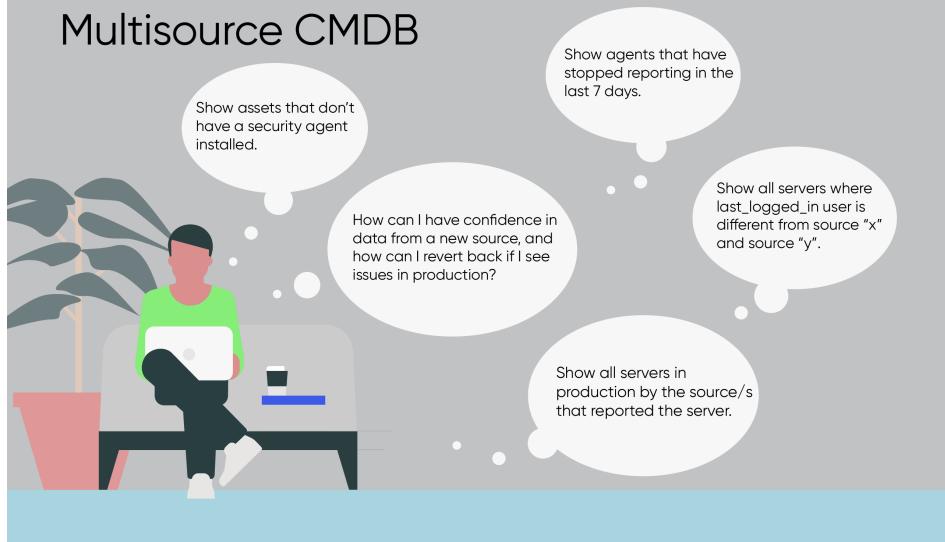
Note: CMDB 360 supports non-CMDB tables. The widely used term Configuration Item (CI), can also refer to a non-CMDB table record. For information about support for non-CMDB tables, see [IRE support for non-CMDB tables](#).

CMDB 360 insights into data source processes



Standardized, unified, reconciled,
and trusted data = **Single Source of Truth**

Multisource CMDB



After data is initially ingested from multiple data sources, several processes are applied to standardize and reconcile the data before it is stored in the CMDB. CMDB 360 provides insights that can help you configure some of these processes.

Using CMDB 360

Use CMDB 360 to:

- Create a dynamic reconciliation rule.
- Control CI updates at the discovery source and CI attribute level.
- Visualize discovery sources of attribute values, at the attribute level.
- Modify reconciliation rules and then recompute CMDB data, reflecting the updated reconciliation rules.
- Revert CMDB data integration from a specific discovery source, if, for example, you realize that the discovery source is not reliable. Recompute CI attribute values, while excluding the discovery source that you want to ignore.
- Validate a new discovery source by comparing its data to data from other discovery sources, which are known to be valid.
- Improve data management, data quality, and operational insights, by querying on CMDB 360 data. Use the CMDB 360 query builder in CMDB Workspace to create queries for CMDB 360 records, discovery sources, and CI records.

Enable and configure CMDB 360

- Activate the ITOM Discovery License (`com.snc.itom.discovery.license`) plugin.
- Navigate to **All > Configuration > CMDB 360 Properties**. Then, in the CMDB 360 Properties pane ensure that the `glide.identification_engine.multisource_enabled` (Enables CMDB 360) property is set to **true**.

By default, CMDB 360 tracks discovery source information for CIs from both CMDB and non-CMDB classes. You can independently enable or disable tracking data for CMDB and for non-CMDB classes, using these system properties:

- `glide.identification_engine.multisource_cmdb_ci_enabled` (Enables capturing CMDB 360 data for CIs from CMDB classes)
- `glide.identification_engine.multisource_non_cmdb_ci_enabled` (Enables capturing CMDB 360 data for CIs from non-CMDB classes)

Report on CMDB 360 data

Use the [CMDB 360 view in CMDB Workspace](#) to gain insights into the CMDB 360 data store. Build reports that, for example, do the followings:

- Find CIs not reported by any discovery source.
- Find discovery sources populating data in your CMDB.
- Compare attribute values across discovery sources.
- Compare attribute values between CMDB and other discovery source.
- Limit reports for CMDB 360 data, to a specific application service, technical service, or a CMDB group.

See [Sample Multisource CMDB queries](#) for more details.

Visualize CMDB 360 data

CMDB 360 is highly verbose in the user interface:

- On the [Reconciliation Rules](#) page in CI Class Manager, click the **Preview Data** tab to see per attribute, discovery sources that are authorized to update that attribute, in precedence order.
- On a CI form, click the **CMDB 360 Data Preview** related link to see per CI attribute, current value in the CMDB and incoming values from other discovery sources.

Logging

Enable logging for CMDB 360 by adding and enabling the system property `glide.cmdb.logger.source.cmdb_multisource`. In the Log [syslog] table, search for entries in which `source="cmdb_multisource"`.

CMDB 360 experience in CMDB Workspace

The CMDB 360 view provides aggregations and analysis of CMDB 360 data which you can use to track activities and identify potential issues of discovery sources. You can also create different types of your own queries and associated schedules and reports to explore CMDB data.

Use the CMDB 360 view in [CMDB Workspace](#) to access all of the CMDB 360 capabilities. For information about all CMDB 360 dashboard settings, see [Configure the CMDB 360 dashboard](#).

Note: Most cards on the CMDB 360 dashboard support non-CMDB tables in their aggregation, or can be configured to provide support. However, the Cls not reported by discovery sources card, for example, doesn't apply to non-CMDB tables. Creating queries for non-CMDB tables is also supported. For information about support for non-CMDB tables, see [IRE support for non-CMDB tables](#).

Access

Requirements:

- Role requirement: sn_cmdb_user (CMDB user) or any role containing sn_cmdb_user
- Additional requirement: [Enable and configure CMDB 360](#)

To access the CMDB 360 view in the CMDB Workspace, navigate to **Workspaces > CMDB Workspace**. In the CMDB Workspace menu bar, select **CMDB 360**.

Potential Issues

Cards on the Potential issues tile show details about Cls with discovery sources that are incorrectly reporting on the Cls.

Clis not reported by discovery sources

Lists Cls that are discovered by multiple discovery sources, but one or more of the discovery sources has stopped reporting within a specified number of days. The [Number of days since Cls were last discovered by a discovery source](#) dashboard setting is used in the card's aggregation.

Drill down on this card to see a list view of the CIs for the card and the specific discovery source per CI that is no longer reporting.

Example: You configure 7 days for the setting. A Linux server CI named backup-linux.sea.com is reported by these discovery sources within the specified number of days:

- ServiceNow - today.
- ServiceWatch - 4 days ago.
- AgentClientCollector - 8 days ago.

In this scenario, the CI shows on the CIs not reported by discovery sources card, since AgentClientCollector reported over seven days ago.

Data mismatch

Lists CIs for which different discovery sources are reporting different values. Attributes are considered mismatched when different discovery sources report different values for the attribute. CIs that appear when you drill down on this card can reveal issues with the individual CI, or your reconciliation rules.

The specific records that appear in the drilled-down list view, depend on the [Data mismatch](#) dashboard settings.

Drill down to the Multisource Data Preview page to see details about all discovery sources with values for the attribute. On the Multisource Data Preview page you can use the Search Attributes box to search for specific attributes and also choose one of the following options:

- **All attributes:** Shows all attributes of the selected class, regardless of whether they have a value in the CMDB.
- **Attributes with CMDB values:** Shows only attributes of the selected class, with a value in the CMDB.
- **Attributes with multisource data:** Shows all attributes of the selected class, and for each attribute shows current CMDB value and any other discovery sources with a value for the attribute.

Regardless of the option that you choose on the Multisource Data Preview page, a discovery source value that was used for the current

CMDB value – is highlighted in green. These values are identical to the CMDB value of the attribute.

Discovery Sources

Cards in the Discovery Sources tile show aggregated counts for your discovery sources.

Number of discovery sources

The total number of discovery sources that report CMDB 360 data.

Total CMDB 360 records

The total number of raw CMDB 360 records in the CMDB 360 data store that contains records for each discovery source report, per each CI attribute.

Reconciled CIs

The total number of unique CIs created in the CMDB after processing incoming data from all discovery sources, including after reconciling data from multiple discovery sources for the same CIs. For more information, see [CMDB Identification and Reconciliation](#).

Discovery source overview

The distribution of CMDB 360 CIs across all reporting discovery sources.

Saved Queries

The Saved queries card shows up to 20 of your CMDB 360 queries. You can use the card to edit and run those queries, or create new queries. Saved queries are sorted on both the card and list view by the most recently created or updated queries.

If your instance has been upgraded from an instance that contained Multisource Report Builder queries, then those queries appear on the Saved Queries card.

- Click a saved query to modify or view it before running.
- Click a query's action icon and then select **Run**.

- Click **View All Queries** to see all the saved queries where you can examine or run a query.
- Click **Create Query** to create a query of one of the following types:
 - **Get Records:** Creates a query that you can use to explore your CMDB 360 data. It queries your CMDB 360 for Cls matching your criteria that are reported by specified discovery sources.
 - **Find Gap:** Creates a query that you can use to analyze gaps in discovery sources reporting your CMDB 360 data. It queries discovery sources that report Cls against discovery sources that don't report those same Cls.
 - **Compare Attribute Values:** Creates a query that you can use to identify Cls with attribute values that differ across multiple discovery sources or against the CMDB. It queries at least two discovery sources and/or the CMDB for Cls that match your criteria.
- **Schedule a CMDB 360 query for a report** to run that query on a regular basis. Scheduling a query enables you to create a CMDB 360 report that integrates the query results with the [platform Reporting feature](#).

Coverage

Cards on the Coverage tile show a breakdown of Cls per the number of discovery sources reporting those Cls.

Clis with a single source

Shows Cls that are only reported by a single discovery source, with a breakdown by the discovery source reporting those Cls.

Clis by number of sources

Shows Cls reported by multiple discovery sources, grouped by the number of discovery sources that are reporting those Cls.

The specific data that this card shows in the list view depends on the '[Coverage cards](#)' dashboard settings.

Configure settings on the CMDB 360 dashboard of the Configuration Management Database (CMDB) Workspace to determine how your CMDB 360 data is analyzed and aggregated. These settings affect the

data that appears on the cards and the records shown when you drill down on those cards on your CMDB 360 dashboard.

Before you begin

Role required: cmdb_ms_admin

Procedure

1. Navigate to **Workspaces > CMDB Workspace**.
2. In the CMDB Workspace menu bar, select **CMDB 360**.
3. Click **Settings**.
4. Configure Global settings.

The **Maximum number of records to process** setting determines the maximum number of records that can show when you drill down on the cards in the CMDB 360 dashboard. If the total number of returned records is greater than the specified setting value, the dashboard trims the list view output according to this setting and the settings of individual cards.

You can use this setting to limit the number of records that CMDB 360 must process. The setting applies to these cards:

- Cls not reported by discovery sources
- Data mismatch
- Cls with a single discovery source
- Cls by number of discovery sources

The CMDB 360 dashboard defaults this value to 100,000.

5. Configure Potential issues settings.

These settings affect the calculations for cards on the CMDB 360 view/Potential Issues tile and the list of Cls that appear when you drill down on those cards.

- a. Configure Cls not reported by discovery sources.

The **Number of days since Cls were last discovered by a discovery source** setting determines the number of days used in the calculation of the Cls not reported by discovery sources card. The card shows Cls that are discoverable by multiple

sources, but at least one discovery source hasn't reported on that CI in the specified number of days.

b. Configure Data mismatch.

These settings affect the calculation of the 'Data mismatch' card and the list of CIs that appear when you drill down the card.

These settings determine which classes to use for the card and the relative weight of each of those classes in the calculations.

Setting	Description
Automatic data weights	Evenly distributes weights between the selected CI classes.
Manual data weights	Specify a custom weight for each CI class.
Select CI classes you want to include in the calculation	Specify the CI classes that you want to check for attribute mismatches. CI classes you specify also include any child classes. Attributes are considered mismatched when different discovery sources report different values for the attribute.
Show CIs where EVERY attribute doesn't match	Select to include only CIs with a mismatch between discovery sources for every attribute that you specify.
Show CIs where ANY attribute doesn't match	Select to only include CIs with a mismatch between discovery sources for any attribute you specify.
Select an attribute	Specify the attributes that you want to check for mismatches.

6. Configure Coverage settings.

These settings affect the calculations for cards on the CMDB 360 view in CMDB Workspace tile and the list of CIs that appear when you drill down on these cards. These settings determine which classes to use for the cards and the relative weight of each of those classes.

Setting	Description
Automatic data weights	Evenly distributes weights for the selected CI classes.
Manual data weights	Specify a custom weight for each CI class.
Select CI classes you want to include in the calculation	Specify the CI classes that you want to include in the CIs with a single source and CIs by number of sources cards. CI classes that you specify also include any child classes.

7. Click **Save**.

Create a Get Records query from the CMDB 360 dashboard of your Configuration Management Database (CMDB) Workspace to help you explore your existing CMDB 360 data.

Before you begin

Role required: sn_cmdb_user and either cmdb_ms_admin or cmdb_ms_editor

Procedure

1. Navigate to **Workspaces > CMDB Workspace**.
2. In the CMDB Workspace menu bar, select **CMDB 360**.
3. On the Saved Queries tile, click **Create Query**.
4. Click **I want to get CMDB 360 data**.

5. Select the CI classes to include in the query.

You can click a selected class to open the condition builder. Use the condition builder to specify conditions that must be met for each class. Use **And** or **Or** to specify multiple conditions.

Select **All Classes** if you want to include all CI classes without conditions.

6. Click **Continue**.

7. Select discovery sources to query on.

The query retrieves CMDB 360 data that originates from the discovery sources you specify.

You can leave the Select discovery sources prompt empty to retrieve data for all discovery sources.

8. Click **Continue**.

9. On the form, select the options:

Results Layout form

Field	Description
Show unique CMDB 360 records	Select if you want to see only unique CMDB 360 records. Records for the same Cls from different discovery sources are consolidated.
Show CI records by discovery source	Select if you want to see records for each CI and discovery source pair.
Limit results to	Limits the query results to Cls that belong to a service or CMDB group. When you select Application Services , Technical Services , or CMDB Groups , a prompt appears. You can use the prompt to specify the service or group that you want the query to filter for.

10. Click **Continue**.

11. Enter a name and description for your query.

12. Click **Save**.

What to do next

Run the query at least once if you want to create a schedule or report.

On the CMDB 360 Query Results page:

- If the number of results exceeds the number of results appearing on the page:
 - Click **Load More Results**: To show the next page of results. The number of results that appear on each result page is specified by the `glide.identification_engine.multisource.query.batch.limit` system property (100 items by default).
 - Click **Load All Results**: To show all results, up to the limit specified by the `glide.identification_engine.multisource.query.max.limit` system property (10000 by default).
 - Click a CMDB 360 Source link to easily access preview data of a source and see more details.
- You can click **Create Schedule** to set up a schedule that runs your query on a regular basis. Scheduling your query enables you to use the query results in reports you create.
- After creating a schedule, you can click **Create Report** to configure a report that you can manage using [Reporting capabilities](#).
- On the Query Results page, access a record to view further details. For example, click a link in the Primary Record column, and then in the CI Details page, click **View CMDB 360 Data**.

Create a Find Gap query from the CMDB 360 dashboard of your Configuration Management Database (CMDB) Workspace to help you find CIs that are not being reported by a discovery source.

Before you begin

Role required: sn_cmdb_user and either cmdb_ms_admin or cmdb_ms_editor

About this task

Gaps in discovery source reporting occur when at least one discovery source reports on the CI and another doesn't, enabling you to identify discovery sources that might have an issue.

Procedure

1. Navigate to **Workspaces > CMDB Workspace**.
 2. In the CMDB Workspace menu bar, select **CMDB 360**.
 3. Click **Create Query**.
 4. Click **I want to find gaps in data between discovery sources**.
 5. Select the CI classes that you want to check for gaps.
 6. Click **Continue**.
 7. Select non-reporting discovery sources.
A non-reporting discovery source is a discovery source that doesn't report CIs as expected. Multiple non-reporting discovery sources have an OR condition with each other.
 8. Select one or two reporting discovery sources.
These discovery sources report the CI as expected and act as a baseline for identifying the gap in reporting.
- You can leave the **Select reporting discovery sources for gap tracking** prompt empty to include all discovery sources in the query.
9. Click **Continue**.
 10. On the form, select the options:

Results Layout form

Field	Description
Show unique CMDB 360 records	Select if you want to see only unique CMDB 360 records. Records for the same CIs from different discovery sources are consolidated.
Show CI records by discovery source	Select if you want to see records for each CI and discovery source pair.
Limit results to	Limits the query results to CIs that belong to a service or CMDB group. When you select Application Services , Technical Services , or CMDB Groups , a prompt appears where you can specify the service or CMDB group that you want the query to filter for.

11. Click **Continue**.
12. Enter a name and description for your query.
13. Click **Save**.

What to do next

Run the query at least once if you want to create a schedule.

On the CMDB 360 Query Results page:

- If the number of results exceeds the number of results appearing on the page:
 - Click **Load More Results**: To show the next page of results. The number of results that appear on each result page is specified by the `glide.identification_engine.multisource.query.batch.limit` system property (100 items by default).

- Click **Load All Results**: To show all results, up to the limit specified by the `glide.identification_engine.multisource.query.max.limit` system property (10000 by default).
- Click a CMDB 360 Source link to easily access preview data of a source and see more details.
- You can click **Create Schedule** to set up a schedule that runs your query on a regular basis. Scheduling your query enables you to use the query results in reports you create.
- After creating a schedule, you can click **Create Report** to configure a report that you can manage using [Reporting capabilities](#).
- On the Query Results page, access a record to view further details.

Create a Compare Attribute Values query from the CMDB 360 dashboard of your Configuration Management Database (CMDB) Workspace to help you find CIs with mismatched attribute values between discovery sources.

Before you begin

Role required: `sn_cmdb_user` and either `cmdb_ms_admin` or `cmdb_ms_editor`.

About this task

The query enables you to determine if there's an issue with how a discovery source reports a CI.

The Compare Attribute Values query compares CIs from different discovery sources for mismatched values. Mismatches occur when a discovery source reports attribute values for a CI that are different from values reported by other discovery sources or the CMDB. You can use the query to identify these CIs and reconcile the attribute values, or fix any issues with the discovery sources.

Procedure

1. Navigate to **Workspaces > CMDB Workspace**.
2. In the CMDB Workspace menu bar, select **CMDB 360**.

3. Click **Create Query**.
4. Click **I want to compare attribute values between discovery sources or against the CMDB**.
5. Select a CI class that you want to compare attribute values for.
You can click a selected class to open the condition builder. Use the condition builder to specify conditions that must be met for each class. Use **And** or **Or** to specify multiple conditions.
6. Click **Continue**.
7. On the form, select the options:

Attributes to compare form

Field	Description
Any attribute doesn't match	Select if you want to retrieve CIs where there's a mismatch with any specified attribute values between the discovery sources.
Every attribute doesn't match	Select if you want to retrieve CIs where there's a mismatch with every specified attribute value between the discovery sources.
Select attributes to compare	Specify the attributes that you want to compare for mismatched values.

8. Click **Continue**.
9. Select your discovery sources that you want to compare attribute values for.

Discovery sources form

Field	Description
Compare to CMDB	Select if you want to compare the specified attribute values against your CI attributes recorded in the CMDB. When

Field	Description
	you compare against the CMDB, you only need one discovery source.
Select discovery sources	The discovery sources that you want to compare. Select at least two.
Limit results to	Limits the query results to CIs that belong to a service or CMDB group. When you select Application Services , Technical Services , or CMDB Groups , a prompt appears. You can use the prompt to specify the service or group that you want the query to filter for.

10. Click **Continue**.

11. Enter a name and description for your query.

12. Click **Save**.

What to do next

Run the query at least once if you want to create a schedule.

On the CMDB 360 Query Results page:

- If the number of results exceeds the number of results appearing on the page:
 - Click **Load More Results**: To show the next page of results. The number of results that appear on each result page is specified by the `glide.identification_engine.multisource.query.batch.limit` system property (100 items by default).
 - Click **Load All Results**: To show all results, up to the limit specified by the `glide.identification_engine.multisource.query.max.limit` system property (10000 by default).

- Click a CMDB 360 Source link to easily access preview data of a source and see more details.
- You can click **Create Schedule** to set up a schedule that runs your query on a regular basis. Scheduling your query enables you to use the query results in reports you create.
- After creating a schedule, you can click **Create Report** to configure a report that you can manage using [Reporting capabilities](#).
- On the Query Results page, access a record to view further details.

Set up a schedule to regularly query for CMDB 360 data. Use scheduled queries to provide CMDB 360 data to reports you create, which can provide insight into how discovery sources populate the Configuration Management Database (CMDB) and the reliability of those discovery sources.

Before you begin

Ensure that you run the CMDB 360 query at least once.

Role required: sn_cmdb_user and either cmdb_ms_admin or cmdb_ms_editor.

Procedure

1. Navigate to **Workspaces > CMDB Workspace**.
2. In the CMDB Workspace menu bar, select **CMDB 360**.
3. On the Saved queries tile, create or access a CMDB 360 query.
If you created a new query, you must run the query at least once before you can click **Create Schedule** on the query results page.
4. Select **Schedule query** on the Results Layout page of the query.
To create a schedule for the Compare attributes values query, select **Schedule query** on the Discovery Sources page.
5. Specify a **Run** frequency and time you want to schedule the query to run.
When you select Weekly or Monthly, you must also select a day of the week or calendar day, respectively.
6. Click **Save**.

What to do next

Create a CMDB 360 report to integrate CMDB 360 query results with platform [Reporting capabilities](#). Each run of the query automatically updates the generated report.

Related tasks

- [Create a CMDB 360 Get Records query](#)
- [Create a CMDB 360 Find Gap query](#)
- [Create a CMDB 360 Compare Attribute Values query](#)

Recompute CI attribute values

Modify reconciliation rules, or exclude a discovery source which is found to be invalid. Then, use the updated reconciliation rules in recomputing CI attribute values, for which those reconciliation rules or discovery source are applicable to.

Before you begin

[Enable and configure CMDB 360](#).

If you want to recompute to apply updated reconciliation rules, then you must first update the reconciliation rules.

Role required: `itil_admin` or `sn_cmdb_admin`

About this task

CMDB 360 automatically generates a recompute task for each recompute that you submit. If you submit multiple recomputes, a recompute task is generated for each operation, but only one task runs at any given time. To list all recompute tasks, enter `cmdb_multisource_recomp_task.list` in the left navigation search box.

There is a maximum number of records that can be included in a single recompute operation. This number is specified by the system property `glide.identification_engine.multisource.recompute.max.ci.limit` (100,000 by default).

Note:

- Recompute skips CIs which are reported by multiple discovery sources, but with different class names.
- Recomputing CI attribute values is not supported with non-CMDB tables.

Procedure

1. Navigate to **All > Configuration > CI Class Manager**.
2. Select a class from the CI Classes hierarchy list.
3. In the left-side pane, expand **Class Info**, and select **Reconciliation Rules**.
4. On the Reconciliation Rules page, click **Recompute**.
5. Select the **Recompute Type**.

•

Replace values from the specified discovery source with value from the discovery source next in priority, according to reconciliation rules: Recompute attribute values for the class CIs, applying priorities specified in reconciliation rules and while excluding the specified **Discovery Source**. Records for the discovery source you are excluding, are also removed from the CMDB 360 data store.

This operation applies to data that exists in the CMDB. If reconciliation rules remain in effect for the discovery source that you have excluded, then future data from that discovery source, can populate the CMDB.

- **Apply updated reconciliation rules:** Recompute attribute values for the class CIs, applying the updated reconciliation rules which are now in effect.

6. Select the **Recompute Scope**.

The scopes grow from one option to the next:

- **Recompute only Cls of this class:** Basic scope of Cls to recompute.
 - **Recompute Cls of this and derived classes:** Expand the basic scope to include Cls from derived classes.
 - **Recompute Cls of this and derived classes, along with selected related items:** Expand the previous scope to also include Cls from specified related items. Select the related items to include in the recompute.
7. Select the **Delete action** for Cls for which the excluded discovery source, is the only discovery source.
- **Delete record:** Delete the CI record from CMDB.
 - **Set record attributes to custom value:** Set a specified CI attribute to a custom value to remove the CI from regular operations without deleting the CI. For example, set the Operational status attribute to Retired.
8. Click **Next** and on the Review page, carefully review the counts for the affected Cls to ensure that all record counts are as you expect. Click **Back** to adjust any settings for the recompute.
9. Click **Recompute**.

What to do next

You can do any of the followings:

- In the status message that appears for the recompute operation, click the link to see the CMDB 360 Recompute Task for more details. The Recompute Task shows the progress and status of the recompute operation.

You can abort the recompute by setting **Status** to **Closed Incomplete** and selecting **Update**. You can then set **Status** back to **Work in progress** to resume recompute from where it was aborted.

- Enter `cmdb_multisource_recomp_task.list` in the left navigation search box, to see the status and progress of all recompute tasks.

Multisource Report Builder (legacy)

Improve CMDB data management by querying and reporting on Multisource CMDB data. Use the Multisource Report Builder to gain insights about how discovery sources are populating the CMDB and their reliability. You can then adjust reconciliation rules to improve the quality of CMDB data, if needed.

CMDB 360 in CMDB Workspace

Starting with the Vancouver release, the Multisource CMDB feature is part of the CMDB 360 feature which is accessible in the CMDB Workspace. Create, view, modify, schedule, create reports, and run CMDB 360 queries using the [CMDB 360 query builder](#) in the [CMDB Workspace](#) store app. Use the CMDB 360 query builder to create queries of the following types:

- [Get Records](#): Queries your discovery sources for CIs that match your criteria.
- [Find Gap](#): Queries for gaps in discovery sources reporting your CMDB 360 data. Queries discovery sources that report CIs against discovery sources that don't report those same CIs.
- [Compare Attribute Values](#): Queries for CIs with attribute values that differ across multiple discovery sources or against the CMDB. Queries at least two discovery sources and/or the CMDB for CIs that match your criteria.

Legacy Multisource Report Builder

Instead of using the CMDB 360 query builder in CMDB workspace, you can still use the legacy Multisource Report Builder as described in this topic.

After you create a Multisource query in the Multisource Report Builder, you can run the query to see the results. You can then also create a Multisource report that integrates the Multisource query results with the platform [Reporting](#) capabilities. High level steps for creating a Multisource report:

1. Create a query, then save and run it.
2. Create a schedule for the query.

3. Create a Multisource report that is based on the Multisource query.

You can create queries that find:

- All the discovery sources populating data in your CMDB.
- CIs not reported by any discovery source.
- All CIs discovered by one discovery source, but not by another discovery source.

You can create other queries that show differences in CI attribute values between multiple data sources, while being compared to the CMDB:

- Show how an attribute value is different between a discovery source and the current CMDB record. For example, find Hardware CIs with different location than what SCCM reports.
- Show how an attribute is different between SourceA, SourceB, and SourceC. For example, show all Computer CIs where RAM is different between SCCM, ServiceWatch, and CMDB.

You can show query results by CI records, Multisource CMDB data records, or discovery sources. You can also limit the report results to CIs within a specific application service, technical service, or a CMDB group.

Create a Multisource query in the legacy Multisource Report Builder

Query the Multisource CMDB data to gain insights about how discovery sources are populating the CMDB, and then use that query to create a Multisource data report.

Before you begin

Enable and configure CMDB 360.

Role required: cmdb_ms_editor

About this task

The Multisource Report Builder page updates dynamically as you set fields. Therefore, some of the fields that are described in the steps below might not appear.

Procedure

1. Navigate to **All > Configuration > Multisource Report Builder**.
2. On the Multisource Report Builder page, select a query to edit or run, or click **New**.
3. Enter **Name** and **Description** for the query.
4. Select the **Result type** for the query.
 - **CI records:** Results show unique CIs from the Multisource CMDB data store.
 - **Multisource data records:** Results show all entries of CI/discovery source combinations from Multisource CMDB data.
 - **Discovery sources:** Results are grouped by discovery sources that match the query criteria.
5. Select **Only show difference** to show differences in CI attribute values and then select the **Type of difference**.
 - **Between CMDB record and discovery source:** Show differences in attribute values between the CMDB data and the specified discovery sources.
 - **Between discovery sources:** Show differences in attribute values between specified discovery sources based on Multisource data.
6. Select the **Class** to apply the query to, or select **All Classes** to apply the query to all classes.
Use the condition builder to specify conditions that must be met for the class. Use **And** or **Or** to specify multiple conditions.
7. Use the slushbucket to select one or more **Discovery source** items to query on.

8. Set **Field to compare** to the class attributes for which to show differences, use the OR and AND operators to compare based on multiple attributes.
The list of attributes is a pre-populated subset of the class attributes, to which you cannot add or remove items.
9. Set **Limit results to** to limit the query results to CIs that belong to a specific application service, technical service, or a CMDB group.
10. Click **Save** and then click **Run**.

What to do next

- On the CMDB Multisource Query Results page:
 - If the number of results exceeds the number of results appearing on the page:
 - Click **Load More Results**: To show the next page of results. The number of results that appear on each result page is specified by the `glide.identification_engine.multisource.query.batch.limit` system property (100 items by default).
 - Click **Load All Results**: To show all results, up to the limit specified by the `glide.identification_engine.multisource.query.max.limit` system property (10000 by default).
 - Click a Multisource CMDB value link or a CI value link to access the respective records and see more details.
 - In the Configuration Item column, click a CI link to open the CI form. In the Related Links section on the CI form, click the **Multisource Data** tab to show Multisource data, such as discovery sources, related to the CI.
- Create a schedule for the query and ensure that the schedule runs at least once. This step is required for creating a Multisource report.

Note: Initially, after creating a query, both the **Create Schedule** and the **Create Report** buttons are grayed out. Only after saving the query, you can create and run a schedule for the query, and only then you can create a report that is based on the query.

Schedule a Multisource query

After saving and running a Multisource query, create a schedule for the query to run automatically on a set schedule. Query results are stored in a results table and you can configure email addresses for the results to be sent to or include the results in CMDB dashboards.

Before you begin

The query for the schedule must be already saved and run at least one time.

Role required: cmdb_ms_user

Procedure

1. Navigate to **All > Configuration > Multisource Report Builder**.
2. On the Multisource Report Builder page, select the query that you want to create a schedule for.
3. On the Multisource Report form, click **Create Schedule**. Then, on the Scheduled Email of Multisource Report Builders, click **New**.
4. Fill in the Scheduled Email of Multisource Report Builders form.

Field	Description
Query	Saved query that was created in Multisource Report Builder.
Users	Users to email the query results to.
Groups	User groups to email the query results to.
Run	Frequency and time to run the query automatically.
Time	

Field	Description
	When you set Run to On Demand , the query runs only when you run it manually.
Email addresses	More ad-hoc email addresses to email the query results to.
Subject	Text that will appear as the subject of the email with the query results.
Introductory message	Text that is included in the body of the email with the query results.
Type	Type of the file with the query results, which will be attached to the email.
Zip output	Enables zipping of the results file.
Conditional	Enable a condition for running the query and specify the condition in the Condition field. If the specified condition isn't met, the query doesn't run.
Condition	<p>Condition that must be met for the query to run (Java script).</p> <p>Appears only if Conditional is selected.</p>
Omit if no records	Disable sending an email for a query run that returns no results.

Note: When using [update sets](#) to port Multisource schedules from a non-production to a production environment, check the **Users** and **Groups** settings in the schedule. Any user or group that doesn't exist in the production environment, and which needs to receive the query results, must be re-added in either of the following ways:

- Manually created in the production environment. In this case, you must also remove the invalid user or group in the production environment (ported from non-production environment) from any schedules and add the new user or group instead.
- Explicitly ported from the non-production to production environment.

5. Click **Submit**.

What to do next

- If **Run** is set to **On Demand** in a schedule, or if you need to run a query randomly even if it has a recurring schedule, you can manually run that query as follows:
 1. Navigate to **All > Configuration > Multisource Report Schedules**.
 2. In the Scheduled Email of Multisource Report Builders list view, select the query that you want to run.
 3. On the Scheduled Email of Multisource Report Builder form, click **Execute Now**.
- Create a Multisource report for the query, that integrates the Multisource query results with the platform [Reporting](#) feature.

Create a report based on a Multisource (CMDB 360) query

After creating, saving, running, and scheduling a Multisource (CMDB 360) query, you can create a Multisource (CMDB 360) report that integrates the query results with the platform Reporting feature. You can for example, include such Multisource (CMDB 360) report in the platform CMDB dashboards.

Before you begin

The Multisource (CMDB 360) query for the report must be already saved, have a schedule, and must have already run at least once.

Role required: cmdb_ms_user

About this task

Creating a report that is based on a Multisource (CMDB 360) query, creates a report source which you can then manage using [Reporting](#) capabilities.

Note: If you are using the [CMDB 360 view in CMDB Workspace](#) to generate the CMDB 360 query and the report, you can skip to step 4 in the procedure below.

Procedure

1. Navigate to **All > Configuration > Multisource Report Builder**.
2. In the Multisource Report Builder list view, select the query that you want to create a schedule for.
3. On the Multisource Report form, click **Create Report**.
4. On the Create a report form, click **Save** or **Run**.

A report shows the results for the most recent query run. If meanwhile the query has changed, then the report shows results that are out of synchronization with the query. When you update the query, ensure to immediately run the updated query so that the report is synchronized with the query.

What to do next

To add a Multisource (CMDB 360) report to the CMDB Correctness Dashboard for example, see [Add a report to a dashboard](#).

Sample Multisource CMDB queries

Use sample queries to create your own Multisource CMDB queries.

Discrepancy in multiple attributes between multiple discovery sources

Field	Setting
Name	Discrepancy in multiple attributes between multiple discovery sources (discovery source vs. discovery source)
Description	Find Linux servers with name containing “backup” which have discrepancy in Disk Capacity OR CPU Count OR Serial Number between discovery sources ServiceNow/ServiceWatch/SCCM/Tivoli.
Result type	Multisource data records
Only show difference	Selected
Type of difference	Between discovery sources
Class	Linux Server [cmdb_ci_linux_server]
Conditions	[Name] [contains] [backup]
Discovery Source	ServiceNow/ServiceWatch/SCCM/Tivoli
Field to compare	Disk Capacity OR CPU Count OR Serial Number
Limit results to	All

Discrepancy in multiple attributes between CMDB record and discovery sources

Field	Setting
Name	Discrepancy in multiple attributes between multiple discovery sources (CMDB vs. discovery sources)
Description	Find Linux Servers with name containing "backup" which have discrepancy in Disk Capacity AND CPU Count AND Fully Qualified Domain Name for discovery sources ServiceNow/ServiceWatch/SCCM
Result type	Multisource data records
Only show difference	Selected
Type of difference	Between CMDB record and discovery source
Class	Linux Server [cmdb_ci_linux_server]
Conditions	[Name] [contains] [backup]
Discovery Source	ServiceNow/ServiceWatch/SCCM
Field to compare	Disk Capacity AND CPU Count AND Fully Qualified Domain Name
Limit results to	All

Servers discovered by ServiceNow but not by Tivoli

Field	Setting
Name	Missing Discovery by Tivoli

Field	Setting
Description	Servers discovered by ServiceNow but not by Tivoli
Result type	CI records
Class	Server [cmdb_ci_server]
Discovery Source	[is] [ServiceNow] [is not] [Tivoli]
Limit results to	All

All discovery sources for backup servers

Field	Setting
Name	Discovery Sources Backup Servers
Description	All discovery sources for backup servers
Result type	Data sources
Class	Server [cmdb_ci_server] and class condition: [Host name][starts with][backup]
Limit results to	All

All Multisource CMDB records where the reported value of Location is different between Altiris and Tivoli discovery sources

Field	Setting
Name	Compare Location-Altiris vs. Tivoli
Description	List all Multisource CMDB records where the reported value of Location is different between Altiris and Tivoli discovery sources
Result type	Multisource data records
Only show difference	Selected
Type of difference	Between discovery sources
Discovery Source	<ul style="list-style-type: none"> • Altiris • Tivoli
Field to compare	Location
Limit results to	All

All Multisource CMDB records for Linux Server, where the Location value is different than the reported value by Tivoli

Field	Setting
Name	Linux Server Location - Diff than Tivoli value
Description	All Multisource CMDB records for Linux Server, where the Location value is different than the value reported by Tivoli.
Result type	Multisource data records

Field	Setting
Class	Linux Server
Only show difference	Selected
Type of difference	Between CMDB record and discovery source
Discovery Source	[is][Tivoli]
Field to compare	Location
Limit results to	All

Components related to CMDB 360

Several types of components are related to CMDB 360 (included in the com.snc.cmdb plugin), such as tables and properties.

Properties

Open the CMDB 360 Properties page by navigating to **All > Configuration > CMDB 360 Properties**. You can hover over the '?' icon for a property, to show property names.

You must have the cmdb_ms_admin role to modify property values.

Property	Description
glide.identification_engine.multisource_enabled	<p>Enables CMDB 360.</p> <ul style="list-style-type: none">• Type: true false• Default value: false• Location: CMDB 360 Properties page

Property	Description
glide.identification_engine.multisource_cmdb_ci_enabled	<p>Enables capturing CMDB 360 data for CIs from CMDB classes (derived from the cmdb_ci class).</p> <ul style="list-style-type: none"> • Type: true false • Default value: true • Location: CMDB 360 Properties page
glide.identification_engine.multisource_non_cmdb_ci_enabled	<p>Enables capturing CMDB 360 data for CIs from non-CMDB classes (not derived from the cmdb_ci class). For example, the Serial Number [cmdb_serial_number] class, or the Software instance [cmdb_software_instance] class.</p> <ul style="list-style-type: none"> • Type: true false • Default value: true • Location: CMDB 360 Properties page
glide.identification_engine.multisource.query.batch.limit	<p>Max number of items to show per query results page, in the CMDB 360 Report Builder. Changing the value of this property, might affect performance when running a query.</p> <ul style="list-style-type: none"> • Type: numeric • Default value: 100

Property	Description
	<ul style="list-style-type: none"> Location: CMDB 360 Properties page
glide.identification_engine.multisource.query.max.limit	<p>Max number of query results to show when you click Load All Results in the CMDB 360 Report Builder. Changing the value of this property, might affect performance when running a query.</p> <ul style="list-style-type: none"> Type: numeric Default value: 10000 Location: CMDB 360 Properties page
glide.identification_engine.multisource.recompute.max.ci.limit	<p>Max number of CIs that can be included in a CMDB 360 recompute operation.</p> <ul style="list-style-type: none"> Type: numeric Default value: 100000 Location: CMDB 360 Properties page
glide.cmdb.logger.source.cmdb_multisource	<p>Enable logging for CMDB 360. CMDB 360 logs are stored in the Log [syslog] table with source set to "cmdb_multisource".</p> <ul style="list-style-type: none"> Type: string

Property	Description
	<ul style="list-style-type: none">• Values: info, warn, error, debug, or debugVerbose• Location: Add to System Properties [sys_properties] table.

Tables

Table	Description
CMDB 360 Data [cmdb_multisource_data]	CMDB 360 data store. Contains the raw data sent by all discovery sources.
CMDB MultiSource Column Metadata [cmdb_multisource_column_metadata]	Mapping of attributes for each class to floatable columns. Used to improves performance of queries that involve high volumes of data.
CMDB Multisource Queries [cmdb_multisource_query]	CMDB 360 query definitions created by the user in CMDB Workspace or in the legacy Multisource Report Builder.
Query Status [cmdb_multisource_query_status]	State of execution, of queries created in CMDB Workspace or in the legacy Multisource Report Builder.
CMDB Multisource Query Results [cmdb_multisource_query_result]	Results for queries created in CMDB Workspace or in the legacy Multisource Report Builder, configured with result type of CI records.

Table	Description
CMDB Multisource Query Result Multisource Records [cmdb_multisource_query_result_ms_record]	Results for queries created in CMDB Workspace or in the legacy Multisource Report Builder, configured with result type of CMDB 360 records.
CMDB Multisource Query Result Discovery Sources [cmdb_multisource_query_result_disco_source]	Results for queries created in CMDB Workspace or in the legacy Multisource Report Builder, configured with result type of Discovery source records.
CMDB MultiSource Recompute Task CIs [cmdb_multisource_recomp_task_ci]	All CIs that are involved in a recompute operation.
CMDB Multisource Recompute Tasks [cmdb_multisource_recomp_task]	Recomputation requests and progress status.

Roles

Role	Description
cmdb_ms_read	Can access and run a CMDB 360 query but can't create a query. Contains cmdb_read role.
cmdb_ms_editor	Can create and run a query, has full read and write access, but can't do Recompute. Contains cmdb_ms_read role.

Role	Description
cmdb_ms_admin	Can create and run a query, and can modify CMDB 360 properties. Contains cmdb_ms_write role.

CMDB data management

The integrity and health of the CMDB is essential for the various features that depend on the data in the CMDB. As the CMDB grows and infrastructures change, the CMDB can accumulate stale, duplicate, or outdated CIs and therefore no longer accurately reflect the IT infrastructure and applications in the organization.

The CMDB Data Manager is an essential tool where you can create, publish, and manage policies that reflect organizational needs for data management. The CMDB Data Manager is a comprehensive and integrated solution which scales to large CMDBs and supports bulk management of CI life cycle operations such as deletion and archival. Use the CMDB Data Manager to automate and govern CI life cycle operations to help maintain the CMDB in a healthy and efficient operational state.

Attesting CIs and managing duplicate CIs are also important data management tasks that help maintain the health of the CMDB.

CMDB Data Manager

CMDB Data Manager is a policy-driven framework for bulk management of CI life cycle operations such as deletion and archival. The CMDB Data Manager is a comprehensive and integrated solution which scales to large CMDBs and copes with rapid changes in a cloud-based world.

Large CMDBs can over time accumulate large amounts of stale CIs which can impact overall performance. Custom mitigating solutions can be difficult to develop and to maintain, and are also prone to errors. The CMDB Data Manager is the tool where you can create, publish, and manage policies. Create policies to automate and govern CI life cycle operations to help maintain the CMDB in a healthy and efficient operational state.

Terms

Policy

A CMDB Data Manager policy captures the overall management plan for a specific life cycle event, such as CI retirement. A policy is associated with a specific subflow (the policy subflow) which creates the tasks (the policy tasks) for the target CIs of the policy. A policy is configured with a policy type and the policy tasks perform operations associated with that policy type, such as archiving or deleting a CI record. Also, you can configure a policy to require an approval.

The policy type, policy subflow, and the policy tasks, are all aligned to a specific life cycle event of CIs. For example, a policy set with the delete policy type, is associated with the delete subflow, and its policy tasks handle the deletion of CIs.

A daily scheduled job processes all published CMDB Data Manager policies.

Policy subflow

The policy [subflow](#) contains the underlying logic to process a life cycle event such as retire or delete. If the policy is configured to require approval, then the policy subflow runs only after a policy task is approved.

The base system provides several common subflows, such as delete, archive, and retire, which you can use with policies. You can also create custom subflows that are needed in the organization.

Policy task

A separate task is created and assigned to each unique Managed By Group value within the set of target CIs in a policy. A policy task triggers the policy subflow, tracks the set of target CIs for the task, and handles the approval of the task, if required.

If a policy requires an approval, the policy tasks do not trigger the policy subflow until a member of the group assignment in the Managed by Group attribute of the target CIs, approves the tasks. If a task is rejected or if the Managed by Group attribute is empty, the task is assigned to an administrator who needs to manually intervene to resolve the task.

If a policy isn't configured to require an approval, then the policy tasks are always automatically approved.

CI exclusion list

A set of CIs to which policies of a specific type do not apply.

CMDB Data Manager experience in CMDB Workspace

You can use the [CMDB Workspace](#) landing page and its views to access high level metrics and details of CMDB Data Manager tasks. For example:

- Use the Important actions tile on the landing page to access data attestation and life cycle approval tasks such as reassignment requests and unassigned overdue tasks. Drill down these task cards to see further details about the tasks.
- Use the My work tile on the landing page to access all open attestation tasks which are assigned to you, or to the group assigned in the Managed by Group attribute and which you are a member of. Review and process these attestation tasks by checking the physical existence of IT infrastructure or applications associated with the CIs in the tasks.
- Use the Governance view to access attestation tasks that are assigned to you or to an assignment group that you belong to in accordance with CMDB Data Manager Attestation policies. Overdue tasks appear in a separated **Overdue tasks** list.

For information about reviewing and processing attestation tasks, see [Review CMDB Data Manager Attestation tasks](#).

To access the Governance view, navigate to **Workspaces > CMDB Workspace** and then select **Governance** in the CMDB Workspace menu bar. Access requires the sn_cmdb_admin (CMDB Admin), sn_cmdb_editor (CMDB Editor), or sn_cmdb_user (CMDB User) role.

Now Platform® data archiving

The functionality that the Archive policy type in CMDB Data Manager provides, relies and extends the Now Platform® [data archiving feature](#), applied specifically for CMDB CIs. While processing an Archive policy to archive CMDB CIs, CMDB Data Manager uses components and processes of Now Platform® data archiving in the following ways:

- The Archive Rule [sys_archive] table contains the Now Platform® archive rules including the Archive Configuration Items CMDB archive rule which CMDB Data Manager Archive policies use.
- Data Manager relies on the Archive scheduled job to run (every hour by default) and process CMDB Data Manager archive policies. The Archive scheduled job is stored in the Schedule Item [sys_trigger] table.
- In the Now Platform® table Archive Job Execution Chunks [sys_archive_run_chunk], the Keys attribute contains the sys_ids of the CMDB CIs to be archived (where Rule ID is the CMDB archive rule ID).
-

Archived records are stored in the Now Platform® archive tables, which are prefixed by 'ar_'. In a similar way, the first time that a CMDB archive job runs, it creates an archive table for each CMDB class (prefixed by 'ar_cmdb'). Therefore, that initial CMDB archive task takes longer than subsequent CMDB archive tasks.

For each Data Manager archive policy, the system batches the policy CIs to be archived into batches of 1000 CIs. The sys_archive_run_chunk table contains a record per each of those batches.

CMDB archive tables, such as ar_cmdb_ci_computer, are listed under **All > System Archiving > Archive Tables**.

When using the CMDB Data Manager to archive CIs, you can also directly apply Now Platform® data archiving features, such as [restoring archived records](#) during a CIs retention period.

Life cycle state definitions

Life-cycle rules define the retirement state for classes in your organization and support the transition of CIs through life cycle stages when using the CMDB Data Manager. After retiring CIs, CMDB Data Manager configures the retired CIs according to the retirement definitions specified by life-cycle rules for the CI's class. Using a Retire, Archive, or Delete CMDB Data Manager policies, requires that an active life-cycle rule exist for each targeted class in the policy. You can activate life-cycle rules in the base system to apply the default definitions, customize those rules, or add life-cycle rules for classes needed in your organization.

The life cycle state of a CI affects the CIs visibility and inclusion in ongoing CMDB processes:

- A retired CI isn't excluded from any views or processes such as CMDB Health.
- An archived CI no longer exists in its active table and instead, it is stored in a separate archive table. Archived CIs are no longer visible or included in processes such as list views, maps, and relations formatters. Archived CIs can be retained for a specified retention period before being deleted from the archive table. During that retention period, archived CIs can be manually restored into an active state by using the Now Platform® feature to [restore archived records](#).
- A deleted CI no longer exists in the table it belonged to and there is no way to restore it into an active state. Deleting a CI is an irreversible operation.

For more information about accessing and managing life-cycle rules, see [Life-cycle rules](#).

Configure the environment for CMDB Data Manager

Prepare your environment for using the CMDB Data Manager:

1. Some policy types such as the life cycle policies Retire, Archive, and Delete, require that an active [life cycle rule](#) exists for each targeted class in the policy. This requirement doesn't apply to all policy types. For example, this requirement doesn't apply to the Attestation policy type. If you attempt to create a policy of a policy type for which this requirement applies but isn't met, an error message appears and the operation fails.
2. You can streamline approval of policies by populating the Manage by Group attribute of CIs that you plan to target in policies. Use the CI Class Manager to populate that attribute for an entire class, in a single synchronization operation. For more information about this data synchronization, see [Set the group for a CI or an entire class of CIs](#). If the Managed by Group attribute is not populated for a CI, then the approval process is directed to the administrator.

Use CMDB Data Manager

To open the CMDB Data Manager, navigate to **All > Configuration > CMDB Data Manager**.

The CMDB Data Manager tool lets you centrally create, edit, review, publish, and track Data Manager policies and the tasks generated by the policies. Use the CMDB Data Manager to [create CMDB Data Manager policies](#) that represent your organizational policies for managing the life cycle of CIs. The available policy types let you create policies to:

- Retire all computers without owners which were created more than a year ago ('Retire' policy type).
- Archive all Linux servers in the Seattle data center which have not been updated for 6 months ('Archive' policy type).
- Delete all containers which have not been discovered in the past week ('Delete' policy type).
- Attest all the CIs in a specific location (Attestation policy type).
- Approve cascade-delete, archive, or retire life cycle tasks generated by [dependent CI management](#).
- Delete orphan, stale, or irrelevant records in non-CMDB related tables. The non-CMDB Related tables in the Related Entry [cmdb_related_entry] table have references to CMDB tables. A CI in a related table can, for example, become orphan if the referenced CI in the CMDB is deleted ('Delete CMDB Related Entry' policy type).

The landing page of the CMDB Data Manager provides a dashboard view of policies, excluded CIs, and open policy tasks. Access the open policy tasks that need attention from the management group or from the CMDB Data Manager administrator. Preview those tasks, and then approve or reject to continue the process.

Manage CMDB Data Manager policies:

- [Create a CMDB Data Manager policy](#)
- [Approve or reject a CMDB Data Manager task](#)
- [Review CMDB Data Manager Attestation tasks](#)
- [Manage CI exclusion lists of CMDB Data Manager](#)

Components installed with CMDB Data Manager

Scheduled jobs

Scheduled job	Description
CMDB Data Manager Archive/Delete Policy Processor	<p>Processes all published policies of type Archive and Delete:</p> <ul style="list-style-type: none">Applies policies only to CIs that are already retired.Processes all Archive policies first, and if no errors encountered, continues to process any Delete policies.
CMDB Data Manager Retire Policy Processor	Processes all published policies of type Retire: Applies policies only to CIs that are not retired.
CMDB Data Manager - Stale Task Cleaner	<p>Cleans up stale CMDB Data Manager tasks by setting the task to Closed Cancelled.</p> <p>The <code>cmdb.data.manager.stale.task.life.in.days</code> system property determines the number of days after which a task is considered stale (90 by default). A task becomes stale if:</p> <ul style="list-style-type: none">The task was created at least 90 days (by default) ago and it is still open.The approval requests are older than 90 days (by default) and the task is not yet approved.

Scheduled job	Description
CMDB Data Manager Delete Related Entry Policy Processor	Processes the Delete CMDB Related Entry policy by deleting the specified related tables from the Related Entry [cmdb_related_entry] table.

Tables

Table	Description
CMDB Data Management Policy [cmdb_data_management_policy] [cmdb_data_management_policy]	Details about CMDB Data Manager policies.
CMDB Data Management Policy Executions [cmdb_data_management_policy_execution] [cmdb_data_management_policy_execution]	Execution records corresponding to each policy evaluation.
CMDB Data Management Policy Runtime Attributes [cmdb_data_management_policy_runtime_attributes] [cmdb_data_management_policy_runtime_attributes]	Current policy metadata including status and summary.
CMDB Data Management Task Control [cmdb_data_management_task] [cmdb_data_management_task]	Open policy tasks generated by published policies.
CMDB Data Management Tasks to Cls [cmdb_data_management_task_to_cls] [cmdb_data_management_task_to_cls]	Associations of tasks to Cls.

Table	Description
[cmdb_data_management_task_to_ci]	
Excluded CIs [cmdb_policy_ci_exclusion_list]	Tracks the CIs that are set to be excluded during policy evaluation.
CMDB Policy Types [cmdb_policy_type]	Policy types supported by the CMDB Data Manager.
CMDB Policy Type Categories [cmdb_policy_type_categories]	Associations of policy types to Flow Designer Categories.

Roles

Role title [name]	Description	Contains roles
CMDB Data Manager administrator [data_manager_admin]	<p>Can access all features in the CMDB Data Manager, including:</p> <ul style="list-style-type: none"> • Full access to assigned tasks. • Full access to policies. • Ability to associate subflow categories to policy type. <p>Can create, edit, and delete policies, calculate previews,</p>	<ul style="list-style-type: none"> • task_editor • data_manager_user • cmdb_query_builder_read

Role title [name]	Description	Contains roles
	approve tasks, and manage exclusion lists.	
CMDB Data Manager user [data_manager_user]	<p>Can view CMDB Data Manager policies in read-only mode and calculate previews.</p> <p>Can perform the following tasks:</p> <ul style="list-style-type: none"> • View assigned tasks. • Update, approve, or reject an assigned task. • Add a CI to an exclusion list from their task. 	cmdb_read

System properties

Property	Description
glide.cmdb.data.manager.delete.batch.size	<p>Size of each batch of CIs that is deleted or archived (affects performance optimization).</p> <ul style="list-style-type: none"> • Type: Integer • Default value: 1000
glide.cmdb.data.manager.subflow.timeout	<p>Threshold (in milliseconds) for subflow running time. A subflow that passes this threshold while running, is cancelled.</p> <ul style="list-style-type: none"> • Type: Integer

Property	Description
glide.cmdb.data_manager.default_archive_time	<ul style="list-style-type: none">• Default value: 60,000 (10 minutes)
cmdb.data.manager.stale.task.life.in.days	<p>Number of days that it is still possible to restore archived CIs from archive tables. After the specified number of days pass, archived CIs are permanently deleted from the archive tables.</p> <ul style="list-style-type: none">• Type: Integer• Default value: 120 <p>Number of days after which a task is considered stale and is set to Closed Cancelled by the CMDB Data Manager - Stale Task Cleaner daily scheduled job.</p> <p>Details:</p> <ul style="list-style-type: none">• Type: Integer• Default value: 90

Life-cycle rules

A life-cycle rule specifies the retirement definition for a class, reflecting processes and protocols in your organization. These rules support the transition of CIs through life-cycle stages as implemented by CMDB Data Manager policies. A life-cycle rule is required for each targeted class in a Retire, Archive, or Delete CMDB Data Manager policy.

Predefined life-cycle rules

The base system includes predefined life-cycle rules for key classes such as Hardware [cmdb_ci_hardware] and Application [cmdb_ci_appl], which are stored in the CMDB Retirement Custom Definitions [cmdb_retirement_custom_definitions] table. For example, the predefined rule for the Service [cmdb_ci_service] class defines that for a retired CI, the value of the attributes [operational status], [Phase], and [Status] is **Retired**.

Predefined life-cycle rules are inactive by default, and you must activate a rule that corresponds a targeted class in a Retire, Archive, or Delete CMDB Data Manager policy. Only in an upgraded instance in which the CMDB Data Manager is used, the life-cycle rule for the Configuration Item [cmdb_ci] class is active. In that case, the retirement definitions in that rule, are in effect throughout the entire CMDB hierarchy due to derivation.

You can use the default definition of a predefined life-cycle rule, or customize a rule to reflect practices in your organization. You can also add life-cycle rules for additional classes. However, each CMDB class can be associated with only a single life-cycle rule.

CMDB Data Manager requirement

The Retire, Archive, and the Delete CMDB Data Manager policies require that for each of the classes that the policy applies to, there's a corresponding active life-cycle rule. After retiring a CI, CI's attributes are configured according to the life-cycle rule for the CI's class.

You can use the default definitions in the predefined rules, customize them, or add new definitions for classes that you need. Predefined life-cycle rules are inactive by default, other than in some upgrade situations in which there are existing policies and specific life-cycle rules are needed by those policies.

Derivation across the CMDB hierarchy

Life-cycle rules are derived throughout the CMDB hierarchy in the same way that other rules, such as identification rules, are derived. Child classes extended from a parent class with a life-cycle rule, derive that rule unless there's a life-cycle rule defined at the child class level.

Edit and activate a life-cycle rule

Create, edit, or activate a life-cycle rule to define the retirement state in your organization, for a specific class. Tables that are targeted in life-cycle CMDB Data Manager policies, must be associated with an active life-cycle rule.

Before you begin

Role required:

- data_manager_user has read access
- cmdb_admin and data_manager_admin have full access

About this task

Life-cycle rules in the base system are inactive and you must activate any rule that you want to use with a life-cycle CMDB Data Manager policy. To activate a life-cycle rule, set its Active attribute to **true**.

Each CMDB class can be associated with only a single rule. Life-cycle rules are stored in the CMDB Retirement Custom Definitions [cmdb_retirement_custom_definitions] table.

Procedure

1. Navigate to **Configuration > CMDB Retirement Definitions**.
2. In the list view, select the life-cycle rule/table that you want to edit and then edit or fill out the CMDB Retirement Custom Definition form.

Field	Description
Active	Activates the life-cycle rule.
Table	The CMDB class that the life-cycle rule applies to, including any child classes of the specified class.
Retirement definition	One or more attribute conditions that together define the

Field	Description
	retirement state for the class. You can only use AND clauses in this definition as the OR clause isn't supported.

3. Click **Update**.

Create a CMDB Data Manager policy

Create a CMDB Data Manager policy to automatically process CIs life cycle event such as deletion. Applying consistent and standard life cycle policies to CIs helps maintain the health of the CMDB.

Before you begin

- The life cycle policies Retire, Archive, and Delete, require that an active [life-cycle rule](#) exists for each targeted class in the policy. If you attempt to create a policy of a policy type for which this requirement applies but isn't met, an error message appears and the operation fails.
- Ensure that any custom subflow that you want to associate with a policy, exists.
- To require a review and an approval for a policy task: Ensure that the Managed By Group attribute is populated in target CIs and that the assigned users have the privilege to approve the policy tasks.
- When [Asset Management](#) is activated, check if there is an asset record associated with that CI before retiring the CI. Check the associated asset record, if there is one, to ensure that the asset state (`install_status`) is Retired.

Role required:

- `data_manager_admin`: Full access to policies
- `data_manager_user`: Can read and preview policies

About this task

Specify for each policy a policy type, a life cycle subflow, and a set of CIs to operate on as target CIs.

Set condition filters to specify the initial set of CIs that the policy applies to. You can then further narrow down the initial set of CIs by using a CI exclusion list for the policy type. During the final preview of the policy, or from a policy task, you can select individual CIs to also exclude for the policy type. The policy eventually applies to the resulting set of CIs, after applying all those filters.

Note: CMDB Data Manager limits the number of target CIs per task to 10,000. Therefore, when a task exceeds that number, Data Manager automatically creates as many additional tasks as needed to include all the CIs for the task. For example, if you target 30,000 CIs in an attestation task, Data Manager breaks down that task into three tasks, each targeting 10,000 CIs.

You can create policies of the following types:

- Delete: Use to remove a CI from its current table with no option to restore the CI into an active state.
- Retire: Use to retire a CI while keeping the CI active in list views and in processes such as CMDB Health.
- Attestation: Use to assign and process attestation tasks that verify the existence of actual IT infrastructure and applications that you own. As CIs are continuously ingested into the CMDB from various data sources, attesting CIs helps to ensure the integrity of the CMDB. For more information about using the Attestation policy type, see [Attesting CIs](#).
- Archive: Use to remove a CI from its current table and store the CI in a separate archive table for temporary retention. Archiving a CI excludes the CI from views and from processes such as maps the relations formatter. During the retention period, you can [restore CIs into active state](#). At the end of the retention period, archived CIs are deleted from their archive table.
-

Delete CMDB Related Entry: Use to clean up any irrelevant or stale data from related tables to help keep CMDB data healthy and relevant as the state of referenced CIs change.

Related tables, such as the Serial Number [cmdb_serial_number] table, aren't part of the CMDB hierarchy but still qualify as CMDB data. Related tables don't inherit from the Configuration Item [cmdb_ci] table, but have at least one column that references a CMDB CI. Related tables are specified in the Related Entries [cmdb_related_entry] table.

You can implement your Retire, Delete, and Archive policies so that they follow [Common Service Data Model \(CSDM\)](#) standards where for example, CIs are archived and deleted only when a CI is already in retired state. When you create these life cycle policies, the system applies processes to manage any dependent CIs that might be left behind. For more details about these processes and about ensuring that the feature is enabled, see [Dependent CIs management](#).

For more information about life cycle state definitions, see [CMDB Data Manager](#).

Procedure

1. Navigate to **All > Configuration > CMDB Data Manager**.
2. On the CMDB Data Manager landing page, in the Policies tile, click **View Policies**.
3. In the CMDB Data Manager Policy and Attributes list view, click **New**.
4. Fill out the fields in the different sections on the **Define Policy** tab.

Note: Some fields are applicable only to specific policy types. Therefore some of the following fields, might not appear for the policy type that you choose.

Field (General)	Description
Name	Unique name for the policy.
Task Assignment Group	Group to assign the task to.

Field (General)	Description
Task Due In Days	Due date for completing the policy tasks such as attestation tasks.
Needs Review	<p>Check to require a review and an approval of the policy tasks, by the group assignment in Cls' Managed by Group attribute or by an administrator.</p> <p>Otherwise, all policy tasks are approved automatically.</p>
Policy Type	Life cycle event or data management action, such as Delete or Attestation, that this policy manages, indicating the type of actions to perform on target Cls.
User Group	Group to use as the task assignment group for the Delete CMDB Related Entry policy type. The list is a subset of user groups from the Group [sys_user_group] table, where at least one member has a data_manager_user role.
Apply Retention Time	<p>The length of time for retaining archived Cls in the archive table before they are deleted.</p> <p>During the specified retention period, you can use the Now</p>

Field (General)	Description
	Platform® data archiving feature to restore archived Cls .

Field (Condition Filter)	Description
Related Entry Table	The related table to apply a Delete CMDB Related Entry policy. The list contains related tables from the Related Entry [cmdb_related_entry] table.
Condition Filter	<p>Criteria that Cls must meet to be included for the policy as target Cls.</p> <p>Additional filtering such as a CI exclusion list, can further narrow down the set of target Cls.</p>

Field (Action)	Description
Subflow	<p>A subflow with the actions that will run on the target Cls for the policy.</p> <p>The subflow typically matches the policy type. For example, if Policy type is set to Delete CMDB Related Entry, then set Subflow to Delete Related Entry Configuration Item.</p> <p>Note: The Attestation policy type is not associated with a subflow.</p>

Field (Schedule)	Description
Frequency	How often to run the task.
Start Time	Time to start running the task when it is due to run.

5. (Optional) Click **Run filter** in the Condition Filter section, to see the resulting list of CIs that match the condition filters.
6. Click **Save**.
7. Click **Preview** and wait for an impact analysis for the policy on the **Preview Policy Impact** page to complete.
This analysis estimates the number of CIs that the policy applies to based on the policy filters, any CI exclusion lists, and the life cycle stage of CIs. For example:
 - If the policy type is retired, CIs that meet the policy filters but are already in a retired state, are not targeted for the policy.
 - If the policy type is archive or delete, CIs that meet the policy filters but are not retired, are not targeted for the policy.
8. (Optional) Select CIs in the target CIs list that you want to also exclude for the policy type. Click **Exclude CI** and then click **Recalculate Preview** to recalculate the data on the preview page.
9. Click **Publish** to activate the policy.
Unpublished tasks are saved as draft policies.

Result

After you publish a policy:

-

A daily scheduled job processes the published policy and policy tasks are assigned as set in the policy. If the policy is associated with a subflow, then policy tasks trigger the policy subflow. Policy execution issues are recorded in an error log with notifications sent to the CMDB Data Manager Administrator.

If the policy is configured to require an approval for its tasks, then email notifications are sent to members of the assignment group in the Managed by Group attribute of the CI. If the policy is associated with a subflow, then a policy task triggers the policy subflow only after the task is approved.

- If the policy is associated with a subflow, then after a policy task is complete, the policy subflow closes the task. For an Attestation policy (which is not associated with a subflow), a user must process all CIs in the task and submit the task to close it.
- For Attestation policies, attestation tasks are assigned to users as specified, and those tasks appear in the [CMDB Workspace](#) when those users log in.
- For some policy types, such as Delete, the list of the target CIs is rolled up in a CSV file that is then attached to the task for tracking purposes.
- Stale tasks are set to **Closed Cancelled** by a daily scheduled job. A task becomes stale when it is still open and not approved after at least 90 days. The number of days after which a task is considered stale is determined by the cmdb.data.manager.stale.task.life.in.days system property.

What to do next

- Click **View Open Tasks** in the Open Policy Tasks tile to track the processing of policy tasks in the CMDB Data Management Task Control list view. The Success Percent column shows the percentage of CIs in the task, for which the task is completed. A CI is counted as complete in an archival task only after the archival process has been fully completed for the CI (and isn't counted as complete while the CI is just staged for archival for example).
- Users log in to the [CMDB Workspace](#) to review and process attestation [tasks](#) assigned to them.
- You can open a policy in CMDB Data Manager and click **Deactivate** to temporarily prevent the policy from running.
- Manage CI exclusion lists of CMDB Data Manager.

Related concepts

- [CMDB Data Manager](#)

Approve or reject a CMDB Data Manager task

If you are an authorized approver, you might receive an email notification directing you to review a CMDB Data Manager policy task. Approve the task to continue its processing, or reject the task to send it to the CMDB Data Manager administrator for a more detailed review.

Before you begin

Role required:

- `data_manager_admin`: Full access to policy tasks and can set state to 'Work in progress' of an unassigned task.
- `data_manager_user`: Read access to policy tasks and can approve or reject an assigned policy task.

About this task

Authorized approvers for a task are all users in the assignment group of the task, with the `data_manager_user` or `data_manager_admin` roles.

Procedure

1. Navigate to **All > Configuration > CMDB Data Manager**.
2. On the CMDB Data Manager landing page, in the Open Policy Tasks tile, click **View Open Tasks**.
3. On the CMDB Data Management Task Control list view, click a task to review.
4. Click the **Approvers** tab in the related lists section.
5. In a record where you are the **Approver**, click the **Requested** value in the **State** column. Then, on the Approval form, set **State** to your approval choice such as **Approved** or **Rejected**.
6. Click **Update**.

Result

1. The Approval field of the task is set to **Approved** or **Rejected**.
2. If the task is approved, then the policy subflow is triggered to continue processing the policy tasks for the target Cls.

Manage CI exclusion lists of CMDB Data Manager

Create a CI exclusion list for the various CMDB Data Manager policy types. Policies of a specific policy type will not target Cls in the exclusion list for that policy type.

Before you begin

Role required:

- `data_manager_admin`: Full access to CI exclusion lists
- `data_manager_user`: Read access and can add a CI to a CI exclusion list

Procedure

1. Navigate to **All > Configuration > CMDB Data Manager**.
2. On the CMDB Data Manager landing page, in the Excluded Cls tile, click **View Excluded Cls**.
3. On the Excluded Cls list view, click **Edit Exclusion List** and fill out the Specify Excluded Cls form.

Field	Description
Policy Type	Type of policy from which to exclude the specified Cls.
Condition Filter	Conditions that Cls must meet to be excluded for the specified policy type.

4. Click **Run filter**.

All the CIs that meet the condition filters, appear in the Results section.

5. In the Results list, select any or all of the CIs that you want to exclude from the policies of the respective policy type.
6. In the Excluded CIs section, click the '+' icon to add the selected CIs to the Excluded CIs list.
7. (Optional) In the Excluded CIs list, select CIs that you want to remove from the exclusion list, then click the trash icon.
8. Click **Save**.

Duplicate CIs

When the instance encounters duplicate CIs during identification and reconciliation, it groups each set of duplicate CIs into a de-duplication task for review and remediation.

De-Duplication tasks

De-duplication tasks provide details about the duplication, including a list of all the duplicate CIs. Review the details of each duplicate CI in the task and the data that was used to determine that the CI is a duplicate.

From a de-duplication task, you can run the Duplicate CI Remediator wizard to reconcile a set of duplicate CIs into a single CI, eliminating the duplication.

Duplicate CI Remediator

The Duplicate CI Remediator is a wizard-like tool that you can use to reconcile a set of duplicate CIs associated with a de-duplication task. You can choose one of the duplicate CIs to retain as an active CI, and then decide how to process the rest of the duplicate CIs. The Duplicate CI Remediator lets you set reconciliation options for attributes, relationships, and related items.

For information about using the Duplicate CI Remediator, see [Remediate a de-duplication task](#).

Properties that affect processing of duplicate CIs

During CMDB Identification, processing of sets of duplicate CIs is determined by:

- Property glide.identification_engine.skip_duplicates (true by default).
- Property glide.identification_engine.skip_duplicates.threshold (5 by default).
- Number of duplicate CIs in a set.

For information about how these properties affect the management of duplicate CIs, see [Detecting duplicate CIs](#).

Main CI

The main CI plays an important role in the remediation of duplicate CIs. The main CI is a single CI from a set of duplicate CIs, that remains active while the rest of the duplicate CIs in the set are potentially retired, deleted, or reconciled into the main CI. Using the Duplicate CI Remediator, you can select a main CI for a set of duplicate CIs associated with a de-duplication task.

The duplicate_of attribute in duplicate CIs, is used to store a reference to the main CI. For duplicate CIs which existed in an instance that was upgraded to the New York release or later, the main CI is unknown. After upgrade, duplicate_of for those duplicate CIs is set to 'Unknown', indicating that the CI is a duplicate but the main CI is unknown.

Before remediation, the CIs in a duplicate CIs set are all duplicates of each other. After remediation, a set of duplicate CIs consists of one main CI, and any number of CIs, each considered a duplicate of the main CI. The duplicate_of attribute of the main CI is empty. The duplicate_of attribute for all the rest of the duplicate CIs in the set, is a reference to the main CI of the set.

Restrictions

IRE uses the duplicate_of field internally by populating it as part of the skip duplicate mechanism, and you should restrict manual updates of that field. For more details, see [Detecting duplicate CIs](#).

If you do attempt to modify the value of `duplicate_of` directly on a CI form or by using a script, the following restrictions are enforced to ensure data integrity:

- A CI cannot be its own main CI (you cannot set a CI as a duplicate of itself).
- A CI and its main CI cannot be from different domains.
-

The `duplicate_of` attribute of the main CI cannot reference any CI as its main CI (you cannot set a CI as a duplicate of another duplicate CI to create a chain of duplicate CIs).

•

If you attempt to set a CI as a duplicate of another duplicate CI, then the CI is set as a duplicate of the main CI of the duplicate CI you are trying to set. If the main CI of the duplicate CI you are trying to set is 'Unknown', the operation fails.

Example: Attempt to set a CI as duplicate of another duplicate CI

CIs	Attempted setting	Result (System enforced)
CI1: <code>duplicate_of</code> = empty		CI1: <code>duplicate_of</code> = CI3
CI2: <code>duplicate_of</code> = CI3	CI1: <code>duplicate_of</code> = CI2	CI2: <code>duplicate_of</code> = CI3
CI3: Main CI		CI3: Main CI

If CI2 is a duplicate of 'Unknown', the operation fails.

•

If a main CI becomes a duplicate of another CI, then it can no longer be a main CI. All CIs that were duplicates of that main CI are set as duplicates of the new main CI.

Example: Attempt to set a main CI as duplicate of another CI

CIs	Attempted setting	Result (System enforced)
CI1: duplicate_of = CI4		CI1: duplicate_of = CI5
CI2: duplicate_of = CI4		CI2: duplicate_of = CI5
CI3: duplicate_of = CI4	CI4: duplicate_of = CI5	CI3: duplicate_of = CI5
CI4: Main CI		CI4: duplicate_of = CI5
CI5: duplicate_of = empty		CI5: Main CI

- If a main CI becomes a duplicate of a CI within the same duplicate CI set, then the selected duplicate becomes the main CI in the duplicate CI set. The rest of the duplicate CIs in the set are set as duplicates of the new main CI.

Example: Attempt to set a main CI as duplicate of a CI within the duplicate CIs set

CIs	Attempted setting	Result (System enforced)
CI1: duplicate_of = CI4		CI1: Main CI
CI2: duplicate_of = CI4	CI4: duplicate_of = CI1	CI2: duplicate_of = CI1
		CI3: duplicate_of = CI1

CI(s)	Attempted setting	Result (System enforced)
CI3: duplicate_of = CI4		CI4: duplicate_of = CI1
CI4: Main CI		

- You cannot delete a CI that is the main CI of duplicate CIs. To delete a main CI, you must first disassociate that main CI with all of its duplicate CIs. Either delete all duplicate CIs that are associated with that main CI, or remove the reference to that main CI from all duplicate_of attributes in any duplicate CIs that have it.

- [Review de-duplication tasks](#)

Review details of de-duplication tasks, and then potentially remediate a de-duplication task.

- [Manually create a de-duplication task](#)

Manually create a de-duplication task when it is not automatically created. You can then use the Duplicate CI Remediator to remediate the manually created task.

- [Remediate a de-duplication task](#)

Remediate a de-duplication task by using the Duplicate CI Remediator wizard. Use the wizard to guide you through the duplicate CI reconciliation process or to apply a custom workflow.

- [Manage default related items list](#)

You can add or remove items from the default list of related items which is used globally in the Duplicate CI Remediator for all de-duplication tasks.

- [Properties for duplicate CIs](#)

Use the properties for duplicate CIs to configure how duplicate CIs are processed.

- [Components installed for duplicate CI remediation](#)

Tables installed to support duplicate CI remediation (included in the com.snc.cmdb plugin).

Review details of de-duplication tasks, and then potentially remediate a de-duplication task.

Before you begin

Role required: itil to view, and itil_admin or cmdb_dedup_admin to remediate a de-duplication task.

About this task

If a duplicate CI is a dependent CI, then you can view the details of the dependent relationship, the Depend on CI, and any relation qualifier chain. If the dependent CI has a lookup table, then you can see the details of the respective lookup table.

Procedure

1. Navigate to **All > Configuration > Identification/Reconciliation > De-duplication Tasks**.
2. Select a task.

Remediate Duplicate Task form

Field	Description
Number	Unique task number.
Assigned to	Person who is responsible for resolving the task.
Short description	Description for the task.
	Details describing how the CI was identified as a duplicate.
Work notes	Note: Not available in de-duplication tasks that were created prior to the London release.

Field	Description
	This field also contains user notes about the decisions and steps of resolving the task.
Priority	Task priority.
State	State of the de-duplication task as it progresses through resolution.

3. In the related lists section, click the **Duplicate Audit Results** tab to see the list of duplicate CIs in this task. You can click a CI to display more CI details.

Column	Description
Duplicate CI	Reference to the duplicate CI. Note: This field is a document ID type, which means that it can reference any record on any table. If the referenced CI is deleted as part of resolving duplicate tasks, then this field is empty.
Relationship	For a duplicate CI that is a dependent CI, this field shows the relationship between the duplicate CI and depend on CI.
Depend on CI	If the duplicate CI is a dependent CI, then this field displays the depend on CI.
Discovery source	Discovery method used for the CI.

What to do next

Analyze de-duplication tasks to determine which CIs should remain active and which of the duplicate CIs in the Duplicate Audit Results lists are stale or incorrect. Click **Remediate** to [remediate a de-duplication task](#).

Related concepts

- [Relation qualifier](#)

Manually create a de-duplication task when it is not automatically created. You can then use the Duplicate CI Remediator to remediate the manually created task.

In some situations, duplicate CIs are not automatically detected and de-duplication tasks are not automatically generated. Such situation happens with a class for which identification rules are not defined and the identification engine cannot be applied. However, you still want to reconcile these duplicate CIs by utilizing the Duplicate CI Remediator.

Use the [CMDBDuplicateTaskUtils](#) API to manually create a de-duplication task in which all duplicate CIs are specified. The de-duplication tasks that you create manually and the automatically created tasks, are stored in the same table ([reconcile_duplicate_task]) and are processed in the same manner. Then use the Duplicate CI Remediator to [remediate those de-duplication tasks](#).

Note: You can manually create a de-duplication task only for CMDB CIs and a CI can be specified as a duplicate CI only in a single de-duplication task.

Remediate a de-duplication task by using the Duplicate CI Remediator wizard. Use the wizard to guide you through the duplicate CI reconciliation process or to apply a custom workflow.

Before you begin

Important concepts associated with the Duplicate CI Remediator:

Main CI

The main CI is one of the duplicate CIs that you want to retain as an active CI while potentially retiring or deleting the rest of the duplicate CIs. The first step in the Duplicate CI Remediator is to select a main CI for the remediation process. The Duplicate CI Remediator lets you choose which attribute values, relationships, and related items from the duplicate CIs to reconcile into the main CI. Alternatively, you can choose not to consolidate any data and retain the main CI as it is.

Default related items list

A list of related items that is used globally in the Duplicate CI Remediator for all de-duplication tasks. All items from the default related items list are selected by default to be merged to the main CI, on the **Merge Relationships and Related Items** tab. Adding or removing items from that slushbucket does not affect the default related items list. See [Manage default related items list](#) for more information.

Note:

- Asset related tables are not included in the default related items list and therefore they are not available for merge.
- If a scenario involves an inactive change request, the **Configuration item** field on the Change Request form is cleared. If the current value is a duplicate CI, then it isn't merged with the main CI.

Review [Properties for duplicate CIs](#) for information about important properties that affect how the Duplicate CI Remediator operates. Including the glide.duplicate_ci_remediator.dry_run property that determines if the Duplicate CI Remediator actually updates the CMDB or not.

Role required: itil to read, itil_admin or cmdb_dedup_admin to write

About this task

As you progress through the tabs of the Duplicate CI Remediator, CIs are not updated. All updates are applied only in the final step, after you click **Remediate**.

Note:

- Merging of attributes and related items that are associated with assets is not supported in the Duplicate CI Remediator.
- The Duplicate CI Remediator behaves differently in remediations that involve a large number of duplicate CIs or where the duplicates are serial numbers. For information about those special cases, see the 'Special remediation scenarios' section at the bottom.

Procedure

1. Navigate to **All > Configuration > Identification/Reconciliation > De-duplication Tasks**.
2. Open the de-duplication task that you want to remediate. On the task form, click **Remediate**.
3. In the Remediate dialog box, select either of the following options:

- **Use the Duplicate CI Remediator (Recommended):** Use the wizard to consolidate duplicate CIs according to your configurations and settings. Follow the Duplicate CI Remediator tabs to configure the reconciliation.

Note: This option is not available with non-CMDB tables.

- **Use a custom remediation workflow:** Use an existing CMDB remediation rule or select **Add New** to [create a new one](#).
 - On the CMDB Remediation Rule form, set **Task type** to **Remediate Duplicate Task** and select **Active**.
 - On the Workflow form, set **Table** to **Remediate Duplicate Task [reconcile_duplicate_task]** and **If condition matches** to **None**.
 - Ensure that the associated workflow remediates duplicate CIs.

In the Remediate dialog box, click **Next** to start the workflow and to exit the Duplicate CI Remediator. The Remediate Duplicate Task form appears, where you can update the **State** of the task.

4. On the **Select Main CI** tab in the Duplicate CI Remediator:
 - a. Select the main CI for this reconciliation using either of the following lists of duplicate CIs. For any CI, you can click the **Name** link to display the CI's attributes, or click the **Related Items** link to display the number of related items.
 - **Recommended:** A subset of the **All** list, containing only system recommended main CIs. System recommendations are based on checking the duplicate CIs for the following criteria:
 - CI with most related items.
 - CI with most relationships.
 - Newest discovered CI.
 - Newest updated CI.
 - Oldest created CI.
 - Previous main CI, if one was previously selected.
 - **All:** All duplicate CIs for the de-duplication task.
 - b. Select one of the following options to choose whether to consolidate any attribute values, relationships, or related items from any of the duplicate CIs into the main CI:
 - **Remediate Manually:** Lets you specify which attribute values, relationships, and related items from duplicate CIs to consolidate into the main CI.
 - **Use Main CI:** Retains main CI attribute values, merges relationships, and merges only the default related items.

Skip to step number 7 as this selection skips all configurations other than choosing the action for the duplicate CIs on the **Determine Duplicate CI Actions** page.
5. On the **Merge Attribute Values** tab review inconsistent values of each attribute. For each attribute, choose to retain the main CI's value, or

choose a value from a duplicate CI for the main CI to be set with.
Then click **Next**.

Column	Description
Attribute	Attribute for which there are different values.
Main CI Value	Attribute value in the main CI.
Other Values	The number of unique Attribute values within the duplicate CIs.

To override the main CI attribute value with a duplicate CI value:

- a. Click the **Other Values** link.
- b. In the attribute dialog box, click **Unique Values** to display only unique attribute values, or **All** to display all attribute values including identical values.
- c. Select a value for the main CI **Attribute** to be set with.
You can click **Reset to Original** to undo the selection of a different attribute value for the main CI.
- d. Click **Select**.

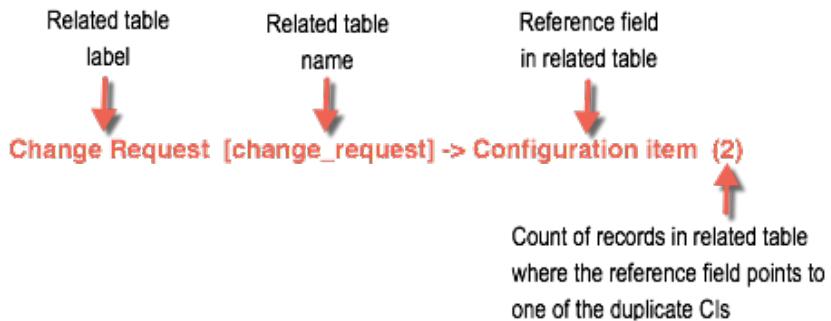
Note: Attributes, such as system fields, discovery fields (discovery_source, last_discovered, first_discovered), and read-only fields (such as the asset field) do not appear in the list.

6. On the **Merge Relationships and Related Items** tab:

- a. In the Merge Relationships section, select whether to merge all relationships from all duplicate CIs into the main CI. Click **View all relationships** to display all the relationships in which a duplicate CI is a parent or a child. You can click a **Parent** or a **Child** link to display more details. Orphan and duplicate relationships of duplicate CIs are deleted if you choose to merge relationships. For more information about the CMDB Health relationship KPI, see [CMDB Health KPIs and metrics](#).

- b. In the Merge Related Items section use the slushbucket to select related items to be merged into the main CI. Click **View all related items** to display all related tables (items) and the count of references in each table to one of the duplicate CIs. You can click the links in **Main CI Related Items** and **Duplicate CIs Related Items** to display details about the related items.

Related items in the slushbucket have the following format:



- By default, all items in the default related items list are selected to be merged.
- Related items (tables) that have no references to a duplicate CI are not listed, unless that table is included in the default related items list.
- Since asset related tables are part of the exclusion list, they are not available for merge.

See [Manage default related items list](#) for more information about configuring a default list of related items.

- c. Click **Next**.

7. On the **Determine Duplicate CI Actions** tab, choose one of the following actions to perform after completing the reconciliation. Then click **Next**.

- **Set attributes to custom values** (recommended): Retain all duplicate CIs. Mark the duplicate CIs as invalid by setting a specific **Attribute** to a specific **Value** for all duplicate CIs. For example, set Operational Status to **Retired** to retire the duplicate CIs.

The `duplicate_of` attribute of the duplicate CIs is automatically set to the appropriate main CI. Also, the duplicate CIs will not be added to any de-duplication task after this task is remediated. If the identification engine is not configured to [skip duplication](#), ensure that identification inclusion rules are configured to exclude the duplicate CIs during identification. This configuration prevents new de-duplication tasks with the same duplicate CIs from being created after remediation.

Note: Discovery fields, system fields, read only fields, and date fields are excluded from the attributes list.

- **Delete:** Delete all duplicate CIs (only the main CI remains).

Note: Review [Roll back and delete recovery](#) for information about reverting the deletion of CIs and related records.

8. On the **Review and Confirm** tab:

- Review the summary of the expected updates for this duplicate CIs reconciliation. Updates are based on your selections and therefore the summary includes only the details that are applicable. This summary can include details of the relationships and related items that will be merged to the main CI, the attribute values that the main CI will be set with, and the number of CIs that will be deleted. Click **Attributes, Relationships, Related Items**, or **Duplicate CI Actions** if applicable, to display further details such as changes to attribute values.
- Click **Remediate** to complete the reconciliation according to your reconciliation settings.
Once complete, the task **State** is set to **Closed Complete**.

Result

The following relationships are deleted without being merged to the main CI:

- Relationships in which the type, child, or parent field is empty.
- Relationships for which merging to the main CI will result in cyclic relationships.

- Relationship for which merging to the main CI will result in duplicate relationships.

What to do next

The reconciliation process runs in the background and may take a while to complete. Upon completion, the system sends a confirmation notice to the remediator of the task. Meanwhile, you can:

- **Review Identification Rules:** Review identification and inclusion rules and make any necessary updates to reduce CI duplication.
- **Check Progress:** View the task activities that are logged as the remediation progresses.
- **View Main CI:** View the main CI for this reconciliation process.

Special remediation scenarios

There are a few special remediation scenarios in which the Duplicate CI Remediator behaves differently.

Large number of duplicate CIs

Support for reconciliation of duplicate CIs in the Duplicate CI Remediator is limited when the number of duplicate CIs exceeds a certain threshold. This threshold is based on the value of the `glide.duplicate_ci_remediator.max.cis` property, which is 1,000 by default. You can update this property to increase the threshold. However, this threshold never exceeds 5,000, even if you set the property to a value greater than 5,000.

When the number of duplicate CIs for a de-duplication task exceeds the threshold, the options available in the wizard are limited:

- On the **Select Main CI** tab, only the **Recommended** list of main CIs appears, and only the **Use Main CI** option is available.
- Recommendations are based only on the oldest created, newest updated, and most recently discovered CIs.
- Reconciliation of attribute conflicts and CI relationships is not supported, and only default related items are reconciled.

Duplicate serial numbers:

The Duplicate CI Remediator is usually applied to duplicate CMDB CIs. However, in some situations de-duplication tasks might be created for duplicate serial numbers. When the Duplicate CI Remediator processes duplicate serial numbers, the merge of relationships from duplicate records, is not referenced and is not applied.

You can add or remove items from the default list of related items which is used globally in the Duplicate CI Remediator for all de-duplication tasks.

Before you begin

Role required: itil_admin

About this task

The default related items list appears on the **Merge Relationships and Related Items** tab in the Duplicate CI Remediator. Globally modifying the list affects all de-duplication tasks being remediated by the Duplicate CI Remediator.

Alternatively, you can modify the list for only a specific task in the Duplicate CI Remediator without affecting the default global list.

Note: Only related items in which the reference field points to Configuration Item [cmdb_ci] in sys_dictionary can be selected for the default related items list. Related items with references to any child of the Configuration Item class cannot be selected for the default related items list, but are still available for merging in the Duplicate CI Remediator for a specific task.

Procedure

1. Navigate to **All > Configuration > Identification/Reconciliation > Duplicate CI Remediator Default Related Items**.
2. On the Default Related Items List for Duplicate CI Remediator page, use the slushbucket to add or remove items from the **Selected** list.
3. Click **Save**.

Use the properties for duplicate CIs to configure how duplicate CIs are processed.

These properties are available for duplicate CIs. To view and edit these properties, the admin role is required.

Properties for duplicate CIs

Property	Description
Attributes in which max_length exceeds this property value (4000 by default) are excluded from the Select Main CI, Merge Attribute Values , and Determine Duplicate CI Actions tabs in the Duplicate CI Remediator wizard. glide.duplicate_ci_remediator.max_field_length	If the max_length for an attribute is equal to the property value, and the size of the data exceeds the property value, then the data is truncated to the property value and the attribute appears in attribute lists. <ul style="list-style-type: none">Type: integerDefault value: 4000Location: Configuration > CMDB Properties > Duplicate CI Remediator Properties <p>Note: This property impacts the performance of de-duplication tasks, therefore be cautious about setting this value.</p>
Comma separated list of related tables in the format '<table>.<reference column>', that are excluded from merging during duplicate CI remediation. glide.duplicate_ci_remediator.related_items_blacklist	<ul style="list-style-type: none">Type: stringDefault value: cert_task.cmdb_ci,cert_audit_result.configuration_item,discovery_log.cmdb_ci,alm_hardware.ci,alm_asset.ci,fm_expense_line.ci

Property	Description
	<ul style="list-style-type: none"> Location: Configuration > CMDB Properties > Duplicate CI Remediator Properties
<p>Threshold for the number of duplicate CIs, which if exceeded, support for reconciliation in the Duplicate CI Remediator is limited (1,000 by default).</p> <p>glide.duplicate_ci_remediator.max_cis</p>	<ul style="list-style-type: none"> Type: integer Default value: 1000 Location: Configuration > CMDB Properties > Duplicate CI Remediator Properties Learn more: See 'Large number of duplicate CIs' in Remediate a de-duplication task. <p>This threshold never exceeds 5,000, even if you set the property to a value greater than 5,000.</p>
<p>Determines whether the Duplicate CI Remediator actually remediates CI duplication by updating records in the CMDB, or not.</p> <p>glide.duplicate_ci_remediator.dry_run</p>	<p>When set to false (default value), updates specified in the wizard are actually performed.</p> <p>You can set this property to true and then test run through the Duplicate CI Remediator without any records actually being updated. In this case, the work notes for the task describe the changes that will happen in an actual remediation.</p> <ul style="list-style-type: none"> Type: true false Default value: false

Property	Description
	<ul style="list-style-type: none">Location: Configuration > CMDB Properties > Duplicate CI Remediator Properties

Tables installed to support duplicate CI remediation (included in the com.snc.cmdb plugin).

Tables installed

Table	Description
Remediate Duplicate Task [reconcile_duplicate_task]	De-duplication tasks.
Duplicate Audit Result [duplicate_audit_result]	CIs associated with each de-duplication task.
Duplicate CI Remediation [cmdb_duplicate_ci_remediation]	Input, selections on each tab, overall status of remediation, and results of each run of the Duplicate CI Remediator.

Attesting CIs

Verify the existence of actual IT infrastructure and applications that you own, systematically and in bulk. As CIs are continuously ingested into the CMDB from various data sources, ensure the integrity of the CMDB. Remove any stale CIs that are associated with IT infrastructure or applications that no longer exists.

Use the [CMDB Data Manager](#) to [create an Attestation policy](#), specifying CIs that need to be attested and the attestation frequency. Assign Attestation tasks to users that are familiar with or that manage the CIs, and who can attest or reject the IT infrastructure or applications that

those CIs represent. Rejected CIs that are no longer needed can then be retired, archived, or deleted from the CMDB.

Users can go to the [Governance view in CMDB Workspace](#) to see their assigned attestation tasks, and then review and process the tasks.

Smart detection and auto-attestation

Smart detection streamlines and simplifies CI attestation. With smart detection you can auto-attest CIs that are automatically detected by discovery programs, based only on recent discovery results.

The following conditions must be met to enable smart detection:

- [Discovery](#) is enabled in your organization or [Service Graph Connectors](#) are implemented.
- The system property `sn_cmdb_ws.attestation.smart_detection.disabled` is set to **false** (default value).

In addition, smart detection uses the following system properties as filters when creating a list of CIs that are candidates for auto-attestation. To be included as candidates for auto-attestation, CIs must be discovered:

- Within the discovery time window specified by the `sn_cmdb_ws.attestation.smart_detection.discovery_window` system property (for example, within the last 30 days)
- By any discovery source that isn't excluded by the `sn_cmdb_ws.attestation.smart_detection.discovery_source.exclusion` system property

Then, when you [review those candidate CIs](#), you can choose to auto-attest them.

Properties associated with Attestation

Property	Description
<code>sn_cmdb_ws.attestation.smart_detection.disabled</code>	Disables smart detection. <ul style="list-style-type: none">• Type: true false

Property	Description
	<ul style="list-style-type: none"> • Default value: false • Values: <ul style="list-style-type: none"> • true: Disable smart detection. • false: Enable smart detection. • Location: Add to System Properties [sys_properties] table. • Learn more: Review CMDB Data Manager Attestation tasks
sn_cmdb_ws.attestation.smart_detection.discovery_source.exclusion	<p>Comma-separated list of discovery sources that are excluded in smart detection processing. For example, a data source that is unreliable in detecting CIs.</p> <p>CIs discovered by discovery sources in the list, can't be candidates for auto-attestation.</p> <ul style="list-style-type: none"> • Type: string • Default value: Manual Entry • Location: System property • Learn more: Review CMDB Data Manager Attestation tasks
sn_cmdb_ws.attestation.smart_detection.discovery_window	Number of days (discovery window) that smart detection uses to determine whether a CI is a candidate for auto-attestation. Only CIs that were discovered within this discovery

Property	Description
	<p>window can be candidates for auto-attestation.</p> <ul style="list-style-type: none">• Type: integer• Default value: 30• Location: System property• Learn more: Review CMDB Data Manager Attestation tasks

Review attestation tasks that are assigned to you or to an assignment group that you belong to in accordance with CMDB Data Manager Attestation policies. Check the physical existence of IT infrastructure or applications associated with CIs in the attestation task and then process those CIs as appropriate.

Before you begin

Role required:

- CMDB administrator: data_manager_admin, sn_cmdb_admin (already includes the data_manager_admin role)
- CI and service owner or manager: data_manager_user, sn_cmdb_user (already includes the data_manager_user role)

About this task

After you receive notifications about attestation tasks assigned to you, check the actual IT infrastructure or applications that you own, and then process the CIs in the task accordingly. Initially, all the CIs in the attestation task are listed in a Not Yet Reviewed list and therefore require attestation. Attested CIs are then moved to an Attested CIs list, while CIs you reject are moved to a separate Rejected CIs list. After processing all the CIs in the task, you can submit and complete the task.

As you process the CIs in the task, details of your activities are captured in the activity stream of the task.

Attesting tasks is performed in the Governance view in the [CMDB Workspace store app](#).

Procedure

1. Navigate to **Workspaces > CMDB Workspace** and then select **Governance** in the CMDB Workspace menu.
Any attestation tasks that are assigned to you or to assignment groups you belong to, appear in the Data Attestation page.
Attestation tasks are associated with details such as the due dates and short descriptions for those tasks. Assignment groups that you are a member of, are also listed.
2. Select a task to review. For a bulk review, select multiple tasks or check the box next to 'Configuration Item' to review all the tasks in the list.
3. Review details of a task:
 - a. (Optional) Administrators can select **Cancel task** if attesting the CIs in the task isn't needed. For example, if the task was created by error. The canceled task is deleted without any further processing.
 - b. (Optional) If the task isn't properly assigned, click **Reassign** and then select one of the following options. Depending on your selection, the task might no longer be assigned to you.
 - **To me:** If the task isn't yet assigned to a specific user and you are the owner of the CIs who can process the task.
 - **To other user:** If you aren't the actual owner of the CIs and there is someone else that is more appropriate as the owner for the task. In the Reassign to other user dialog box, set **Assign to** to that user and enter the reason for the reassignment. Then click **Proceed**.

For a CI owner, only users from the assignment group that the task is currently assigned to, appear. For Admins, all users with the data_manager_user role appear.

- **Send for reassignment:** If you aren't the appropriate user for this task and the administrator needs to assign the task to

someone else. In the Send for reassignment dialog box, enter the reason for the operation and then click **Submit** to send a notification to the administrator.

- c. Click **Review CIs to attest** to continue the CI attestation process. If the task is not assigned to anyone, it is automatically assigned to the current user.
4. (Optional) If the Smart detection dialog box appears, click **Review** to review the CIs or **Auto-attest** to attest the detected CIs without a review. If you click **Cancel**, you can later click **Run auto-attestation** in the Smart detection widget. When reviewing the CIs to attest, the list in the Smart detection pane is filtered to CIs that are candidates for auto-attestation, letting you auto-attest all those CIs in bulk.
 - a. Review the details of the CIs in the list.
 - b. (Optional) Click **Auto-attest** to automatically attest all the CIs in the list and to move them to the Attested CIs list.
 - c. (Optional) Click **Cancel auto-attestation** to leave all the CIs in the Not Yet Reviewed list and to later continue attesting those CIs regardless of smart detection.
5. Review the CIs in the Not Yet Reviewed list. You can click a CI to open a dashboard with more details such as **CI health**, key properties, and **CMDB 360 data**. The details that appear on the CI dashboard depend on settings and activation of different features. Use the following cards on the right hand bar to get help and status:



- Click Attestation actions () to get help on the possible actions you can take while reviewing CIs.
- Click Attestation status () to see your progress in reviewing and attesting CIs.
- Click Smart detection status () to see if any CIs are candidates for auto-attestation and to apply auto-attestation if applicable.

6. In the Details tab of the Attestation Review CIs pane, select the CIs that you are ready to process and select an action to apply to all selected CIs.

•

Select the CIs that you can attest their existence and click **Attest** to move them to the Attested list.

To undo attestation, select the Attested list, select the CIs that you want to move back to the Not Yet Reviewed list and click **Unattest**. Or, select the CIs that you want to move to the Rejected list and click **Reject**.

•

Select the CIs that you can't attest their existence and click **Reject** to move them to the Rejected list. In the Confirm rejecting CIs dialog box, enter an explanation and then click **Proceed**.

To undo rejection, select the Rejected list, select the CIs that you want to move back to the Not Yet Reviewed list and click **Unattest**. Or, select the CIs that you want to move to the Attested list and click **Attest**.

•

Select the CIs that shouldn't be included in this task and in future Data Manager attestation policies, and click **Exclude**. In the Confirm excluding CIs dialog box, enter an explanation and then click **Proceed**.

Those excluded CIs are added to the exclusion list of CMDB Data Manager attestation policies. For more information, see [Manage CI exclusion lists of CMDB Data Manager](#). Only an admin can undo this operation using the CMDB Data Manager.

•

Select any CIs that you identify as duplicates of other CIs and click **Remove duplicates from task**. In the Confirm removing duplicate CIs dialog box, enter an explanation and then click **Proceed**.

Those duplicate CIs are removed from the attestation task and a de-duplication task is generated for them, which is similar to the de-duplication tasks created by the Identification and Reconciliation Engine (IRE). For more details about how to process and remediate de-duplication tasks, see [Duplicate CIs](#).

Note: Only an admin can undo this operation.

- After processing all the CIs in the task, when the Not Yet Reviewed list is empty, the **Submit** button is enabled. Click **Submit**, and then in the Submit attestation dialog box click **Submit** to close the task.

CMDB groups

A CMDB group is a collection of CIs that lets you apply CI actions collectively to all the CIs that are members in the group.

For example, a CMDB CI Lifecycle Management API can use a CMDB group scriptable API to retrieve the group's list of CIs, and then apply a CI Lifecycle Management action collectively to all the CIs. You can also use a CMDB group with the [Dynamic CI Group](#) service population method, to populate an application service.

Group type

A CMDB group is configured with a group type.

- If a CMDB group type is set to Health, then the CIs in the group can be monitored by CMDB Health, and the aggregated health is reported for the group as a whole in the CMDB group view dashboard. For example, you can monitor health only for CIs in a specific location.
- If a CMDB group is set to CMDB Workspace, then that group appears in the [Management view](#) in the [CMDB Workspace](#).

Create and populate a CMDB group

Depending on the group type, you can populate a CMDB group by manually adding individual CIs, selecting saved CMDB queries, or building encoded queries in the CMDB group itself. The resulting CIs from each query are added as members to the group.

Before you begin

Roles required:

- To view CMDB groups - itil
- To use a CMDB queries - cmdb_query_builder on top of itil
- To manually add CIs - itil or asset

Also, to populate a CMDB group using a CMDB query, a saved CMDB query must exist.

Procedure

1. Navigate to **All > Configuration > CMDB Groups**.
2. In the CMDB Groups pane, click **New**.
3. Fill out the form, right-click the title bar and select **Save**.

CMDB Group form

Field	Description
Group Name	A unique name for the group.
Group type	<p>• Default: Basic group type which can be populated by manually adding CIs, saved queries, and encoded queries.</p> <p>• Health: Sets CMDB Health to monitor the health of the group CIs and aggregate health results for the group as a whole.</p> <p>Can be populated only by encoded queries.</p>

Field	Description
	<p>Note: Dynamic filters are not supported when populating this type of CMDB groups.</p> <p>CMDB Workspace: Custom class group which appears in the CI Overview panel in CMDB Workspace store app.</p> <p>Can be populated only by encoded queries.</p>

4. To use saved CMDB queries:

- a. Click **CMDB Group Contains Saved Queries** and then click **Add Query**.

- b. Select a query from the **Query Builder Saved Query** list.

- c. Click **Submit**.

The query that is used returns a list of CIs of the class in the starting node of the query.

5. To manually add CIs:

- a. Click **CMDB Group Contains Configuration Items** and then click **Edit Manual CI**.

- b. (Optional) Add filters.

- c. Select CIs in the **Configuration Item** list and click the '+' icon at the bottom.

- d. In the **Group members** list, select the CIs to add to the group.

- e. Click **Save** or **Save and Exit**.

- f. In the **Save Confirmation** dialog box, click **OK**.

- g. Click **Submit**.
6. To use encoded queries:
 - a. Click **CMDB Group Contains Encoded Queries** and then click **New**.
 - b. Fill out the CMDB Group Contains Encoded Query form with the query conditions that filter the CIs to be included in the group.

Note: Dynamic filters aren't supported for CMDB health-type groups, even though it's possible to add them in a condition clause.

Field	Description
Class	Class for which the encoded query applies to.
CI Overview Condition	<p>Filter that is used in the calculation for the CI Overview chart in CMDB Workspace.</p> <p>Applies only if the <code>sn_cmdb_ws.ci_overview.enable_simple_condition</code> system property is set to true (false by default). Improves performance by yielding less results when there is a large amount of data in an environment that hasn't migrated to CSDM.</p> <p>This condition isn't used when clicking Show All CI.</p>
Condition	Filters the CIs that are included in the group. Used if Simple Condition isn't applicable.

- c. Click **Submit**.

What to do next

Click **Show All CI** to show all CIs from all the result columns of the query. However, only CIs from CMDB tables are shown.

Show CI Lifecycle Management details for CMDB group CIs

Display CI Lifecycle Management operational state and CI actions that apply to the CIs that are members of a CMDB group.

Before you begin

Role required: none

About this task

If the CMDB group is based on a CMDB query, then the query runs in real-time and displays the resulting CIs. If the query does not complete successfully due to timing out or for other reasons, then appropriate error messages are displayed.

Procedure

1. Navigate to **All > Configuration > CMDB Groups**.
2. On the **CMDB Groups** page, click a CMDB group.
3. Click **Show All CI**.

Field	Description
Configuration Item	CI group member.
Class	Class of CI group member.
Operational Status	CI Lifecycle Management operational state of the CI such as 'Repair in Progress' or 'Operational'. Possible operational states are defined in the choice list of the Operational status field in the cmdb_ci table.

Field	Description
Actions	CI Lifecycle Management actions that apply to the CI such as 'Cloning' and 'Provision'. Possible actions are defined in the CMDB CI Actions [statemgmt_cmdb_actions] table.

CMDB Compliance

CMDB Compliance is a tool set that enables administrators to certify CMDB data for correctness and fix any discrepancies found in the data.

Note: For compliance in the context of internal business goals and objectives, and external legislation and regulations, see [GRC: Policy and Compliance Management](#).

Certification options

CMDB Compliance offers these certification options to suit the size and requirements of your organization:

Option	Description
Desired State	Automatically compares the actual attributes and relationships of specific ServiceNow records against the desired states for those records. For example, an audit can detect a Linux database server with insufficient RAM or whose Depends on relationships with another CI is incorrect. The system then publishes any discrepancies found and automatically assigns follow-on tasks to qualified users to bring that server into compliance.

Option	Description
Architecture Compliance	Automatically compares the actual attributes of specific Cls, such as CPU count, RAM, or disk size against the expected attributes for those Cls. The system publishes any discrepancies found and automatically assigns remediation tasks to qualified users.

Compliance Templates and Audits

The Templates and Audits modules on the top level of the Compliance menu enable a certification_admin user to create, edit, and delete all template and audit types.

You can use Compliance Templates and Audits to evaluate records for any table in the ServiceNow system, not just those tables extending the Configuration Item [`cmdb_ci`] table. Compliance audits certify record attributes only. Compliance templates can be used in Control Test Definitions in Governance Risk and Compliance.

Compliance Activation

Compliance functionality is provided by the Certification Core (`com.snc.certification_core`) plugin which contains shared functionality required for certification audits.

The Certification Core (`com.snc.certification_core`) plugin consists of the following plugins, and is activated by default.

- Activated by default: [Desired State Certification](#) (`com.snc.certification_desired_state`)
- [Activate: Architecture Compliance](#) (`com.snc.architecture_compliance`), which automatically activates the Version Management (`com.snc.version`) plugin that manages certification filter and template versions.

- **Activate:** [Data Certification](#) (com.snc.certification_v2), which automatically activates the Version Management (com.snc.version) plugin that manages certification filter versions.

Installed with Compliance

These components are installed with the Certification Core plugin.

Demo data is included with the Desired State and Architecture Compliance plugins.

The Certification Core plugin adds or modifies these tables.

Compliance Certification Core tables

Name	Description
Audit [cert_audit]	Contains all the data required to run an audit, including the users assigned to follow-on tasks and the run schedule.
Audit Result [cert_audit_result]	Contains the results of specific, certification audits.
Follow On Task [cert_follow_on_task]	Contains the tasks that were generated from an audit discrepancy.
Certification Template [cert_template]	Contains the definition of the desired state of the record. The template includes a filter that identifies the records to evaluate and the expected attributes and relationship values. Contains the records to certify, the expected attributes, and the expected relationship values.

Name	Description
Certification Condition [cert_cond]	Base table that defines the desired attribute or relationship conditions used in templates.
Certification Attribute Condition [cert_attr_cond]	Contains the conditions that define the desired CI attribute values. This table extends the Certification Condition [cert_cond] base table.
Certification CI Relationship Condition [cert_ci_rel_cond]	Contains the CI to CI relationship conditions. This table extends the Certification Condition [cert_cond] base table.
Certification User Relationship Condition [cert_user_rel_cond]	Contains the CI to user relationship conditions. This table extends the Certification Condition [cert_cond] base table.
Certification Group Relationship Condition [cert_group_rel_cond]	Contains the CI to group relationship conditions. This table extends the Certification Condition [cert_cond] base table.
Certification Related List Condition [cert_related_list_cond]	Contains the related list conditions. This table extends the Certification Condition [cert_cond] base table.
Certification Filter [cert_filter]	Contains a certification filter, including the table that contains the records to audit and the filter conditions.

User roles

The certification role is automatically assigned to all users with the **itil** role when the [Certification Core plugin](#) is activated or when compliance applications are upgraded. Certification core installs two business rules, both called **Add Certification Role To Manager**, that perform similar tasks on different tables. One rule checks for a manager specified on the

User [sys_user] table, and the other checks for the certification role on the User Role [sys_user_has_role] table. When both a manager and the certification role are specified for a user, the system automatically grants the certification role to the manager. This functionality ensures that a certification task can be escalated successfully to the next level. The system grants this automatic role to the user's immediate manager only and not to others up the management chain.

Note: When a manager has only the certification role and no other role, the manager is considered a Requester and is not counted as a subscribed user (Fulfiller).

Compliance Certification Core user roles

Name	Contains roles	Description
certification	none	Can read and update certification tasks to resolve discrepancies.
certification_filter_admin	none	Can create, read, and update certification filters.
certification_admin	certification, certification_filter_admin	Can manage the entire certification process. These users can create, edit, and delete all certification records.

UI policies

Compliance Certification Core UI policies

Name	Table	Description
Make table read only	Audit [cert_audit]	Sets the table field derived from the selected filter to read-only.

Name	Table	Description
Hide Audit Type	Audit [cert_audit]	Hides the Audit type field.
Hide next scheduled run	Audit [cert_audit]	Hides the Next scheduled run date when an audit is inactive or on-demand.
Show task fields when create tasks is set to true	Audit [cert_audit]	Displays all fields related to creating tasks when the user selects the Create tasks check box.
Make name mandatory	Audit [cert_audit]	Makes Name a mandatory field.
Prevent editing of Last run date	Audit [cert_audit]	Makes Last run date field read-only.
Show User field	Audit [cert_audit]	<p>Shows or hides fields based on the Assignment type selected. The system shows the User field when you select the following assignment types:</p> <ul style="list-style-type: none"> User Field if the Assign to empty option is Create Assigned Task. Specific User
Show Assign to fields	Audit [cert_audit]	Shows or hides fields based on the Assignment type

Name	Table	Description
		selected. The system shows the Assign to field when the assignment type is User Field.
Show Assignment Fields	Audit [cert_audit]	<p>Shows or hides fields based on the Assignment type selected. The system shows the Assign to empty field when you select either of the following assignment types:</p> <ul style="list-style-type: none"> • User Field • Group Field
Show Group field	Audit [cert_audit]	<p>Shows or hides fields based on the Assignment type selected. The system shows the Group field when you select either of the following assignment types:</p> <ul style="list-style-type: none"> • Specific Group • Group Field if the Assign to empty option is Create Assigned Task.
Hide "run" associated fields when active is set to false	Audit [cert_audit]	Hides these scheduling fields when the audit is inactive:

Name	Table	Description
		<ul style="list-style-type: none"> • Run • Day • Time • Last scheduled run
Show script window on Scripted Audit	Audit [cert_audit]	Displays the Run this script field when the audit type is Scripted.
Make table read only	Certification Condition [cert_cond]	Sets the table field derived from the selected filter to read-only.

Script includes

Compliance Certification Core script includes

Name	Description
DesiredStateUtil	Utility functions for desired state, used to clone a template for Insert functionality.
CMDBRElationshipAjax	Tool to get all relationships for a given table.
RelationshipQueryParseAjax	Parses condition filters. This script include is the internal code used in generating the compliance conditions.
CertificationUtils	Utility functions for certification that find Next run time value, and so on.

Name	Description
CertTaskEscalationTimerPercentage	Utility method for setting escalation timer durations.
ConditionUtilsAjax	AJAX utilities for parsing queries into a human-readable format.
DeleteInactiveVersionsAjax	AJAX server-side script to delete all inactive versions of a record.

Client scripts

Compliance Certification Core client scripts

Name	Table	Description
Make audit type read only if not new	Certification Template [cert_template]	Sets the correct audit type for new records, and if the record is not new, sets the Audit type field to read only.
Update table name (filter)	Audit [cert_audit]	Updates the table Name field when the filter is updated.
Update table name	Audit [cert_audit]	Updates the table Name field when the template is updated.
Set table name on new	Audit [cert_audit]	Returns the table name from the template or filter.
Update table name	Certification Template [cert_template]	Updates the table Name field when a new filter is chosen and checks all existing conditions to see if they work for the new table.

Name	Table	Description
Show conditions when table is set	Certification Template [cert_template]	Shows and hides conditions appropriately when the table is set.
Reset filter when audit type changes	Certification Template [cert_template]	Clears the filter and updates the lists shown when the audit type is changed.

Business rules

Compliance Certification Core business rules

Name	Table	Description
Clone condition	Certification Condition [cert_cond]	Part of certification versioning. This business rule retains the original ID when a condition is changed.
Copy audit type from audit	Audit Result [cert_audit_result]	Ensures that all audit results have the same audit type as the audit that generated them.
Copy values from template	Audit [cert_audit]	When a user selects a template, and updates the table, filter, and audit type from the template.
Delete condition	Certification Condition [cert_cond]	Part of certification versioning that deletes a condition.
Prevent deletion of audit with results	Audit [cert_audit]	Prevents deletion of an audit containing results.

Name	Table	Description
Prevent delete of Filter with Template	Certification Filter [cert_filter]	Prevents deletion of a filter still linked to a template or audit.
Prevent deletion of result with task	Audit Result [cert_audit_result]	Prevents deletion of an audit result with an attached task.
Prevent delete of Template with Audit	Certification Template [cert_template]	Prevents deletion of a template still being used by an audit.
Update conditions' tables	Certification Template [cert_template]	When storing template conditions, properly run all workflows and update the condition fields to contain the display version of the conditions.
Update filter version	Certification Filter [cert_filter]	Creates a version when the filter changes in any meaningful way.
Update next run time	Audit [cert_audit]	Updates the time in the Next scheduled run field when an audit is modified.
Update next run time during execution	Audit [cert_audit]	When the audit runs, update the Next scheduled run field to the next time the audit is scheduled to run.
Update table	Certification Template [cert_template]	Update the stored table to the table of the filter.

Name	Table	Description
Update template version	Certification Template [cert_template]	Creates a version when the template changes in any meaningful way.

Compliance Overview module

The Compliance Overview module is a type of a homepage.

About this task

The Compliance Overview module summarizes:

- Current audit states
- Outstanding certification tasks
- Compliance discrepancies
- Upcoming audits
- General state of compliance audits for Data Certification, Desired State, Architectural Compliance, and Scripted audits

Only users with certain roles can access the Overview module. The different levels of access are:

Access levels per role

Role	Access
certification	View (view overview page and refresh reports)
certification_admin	<ul style="list-style-type: none">• View (view overview page and refresh reports)

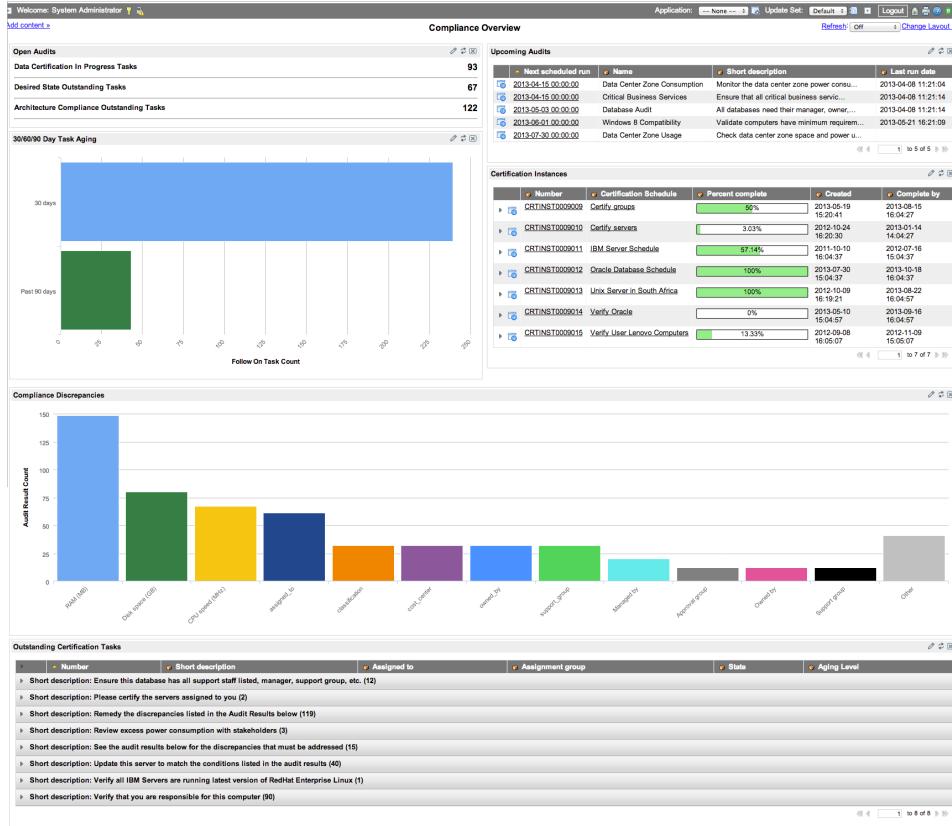
Role	Access
	<ul style="list-style-type: none">Customize (refresh, add, delete, and rearrange reports) <p>View, customize</p>
admin	<ul style="list-style-type: none">View (view overview page and refresh reports)Customize (refresh, add, delete, and rearrange reports)Edit (can edit gauges)

Procedure

1. Navigate to **All > Compliance > Overview**.
2. Click elements within the reports to obtain more information.
For example, click the **Disk space (GB)** bar in the Compliance Discrepancies chart to open a list of audit results filtered by disk space attributes.

Example

Compliance Overview



Architecture Compliance

Architecture Compliance manages scheduled or on-demand audits of CMDB data to determine which configuration items (CI) match expected attributes. The compliance audits check servers to ensure that their physical resources, such as CPU speed or memory, comply with certain standards.

The compliance process checks servers to ensure that their resources, such as CPU speed or memory, comply with standards set by your organization. Audit reports show any discrepancies in the attributes of the target CIs, and ServiceNow automatically assigns follow-on tasks to qualified users who can remediate those discrepancies.

The administrator responsible for compliance checking creates template definitions of expected attributes and then schedules an audit to check Cls for compliance. The audit results identify Cls that pass certification and itemize the discrepancies in those Cls that fail. ServiceNow automatically generates and assigns follow-on tasks to track the process of getting the Cls back into compliance. Users with the admin role activate Architecture Compliance.

Architecture Compliance roles

To access or configure certification elements, a user must have the certification_admin role. These users can create, update, and delete filters if they have the proper access to necessary tables.

In the base system, certification_admin users have limited system rights and do not have access to all the necessary tables. When assigning compliance resources, make sure to grant additional roles to the certification_admin user as needed. For example, the certification administrator needs roles that grant access to these tables:

- Company [`core_company`]
- Cost Center [`cmn_cost_center`]
- Schedule [`cmn_schedule`]

Architecture Compliance Process

Perform these tasks in this order to certify configuration items with Architecture Compliance.

1. Create a filter.

Create a filter that defines a subset of configuration items to certify. You can create multiple versions of a filter, and then activate the version you want to use for compliance checking. Architecture compliance only supports filters on the Configuration Item [`cmdb_ci`] table and all tables that extend it.

2. Create a template.

Create template conditions using values from reference fields in a related list or conditions that define the expected physical attributes

of each CI in an audit. The template uses a filter to determine which configuration items the system examines based on these conditions.

3. Create and run an audit.

Create and schedule an audit or run an audit on demand. The audit generates a set of results based on the conditions in the template you specify.

4. View audit results.

View the audit results which display any discrepancies between the expected state, as expressed by the template conditions, and the actual state of the target configuration items.

5. Correct discrepancies.

Correct the discrepancies the audit found by completing the follow-on tasks created by the system.

The Architecture Compliance Overview module displays various architecture compliance reports. The Overview module is a type of homepage.

Only compliance users with certain roles can access the Overview module. The different levels of access are:

Access levels per role

Role	Access
certification	View (view overview page and refresh reports)
certification_admin	<ul style="list-style-type: none">• View (view overview page and refresh reports)• Customize (refresh, add, delete, and rearrange reports) View, customize

Role	Access
admin	<ul style="list-style-type: none">View (view overview page and refresh reports)Customize (refresh, add, delete, and rearrange reports)Edit (can edit reports)

Using the Architecture Compliance Overview Module

To use the Architecture Compliance Overview module, navigate to **Compliance > Architecture Compliance > Overview** and click elements within the gauges to obtain more information.

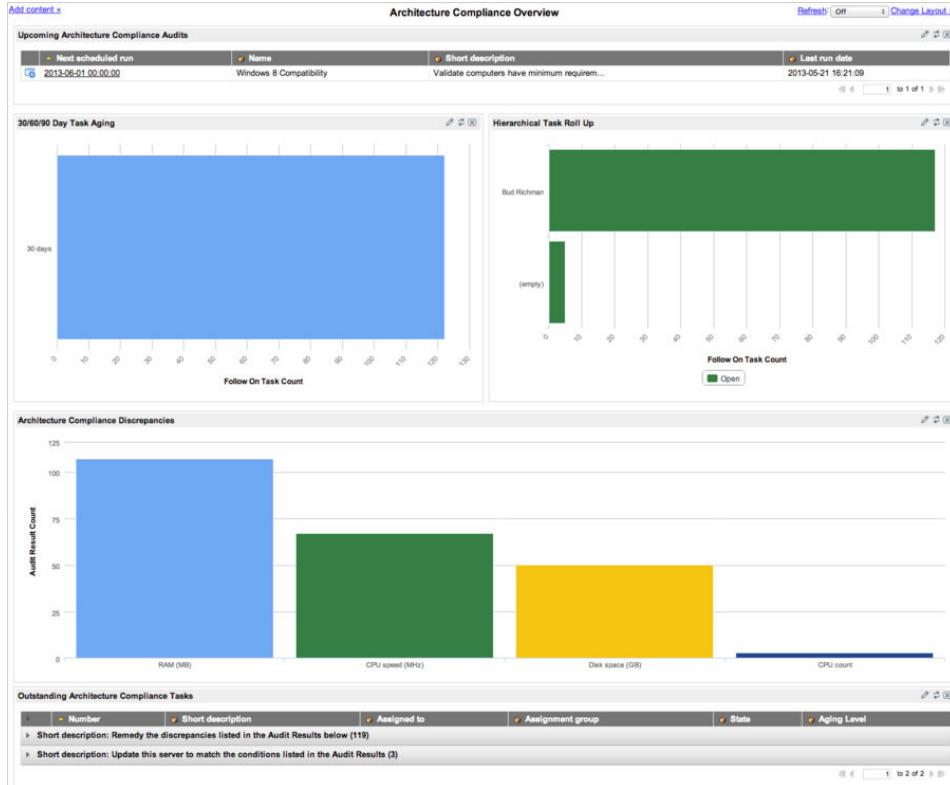
The available reports are:

Architecture Compliance Overview Module Gauges

Report	Description	Table
30/60/90 Day Task Aging	All outstanding follow-on tasks grouped by age in 30-day increments	Certification Task
Architecture Compliance Discrepancies	All audited attribute discrepancies	Audit Results
Hierarchical Task Roll Up	All follow-on tasks grouped by Assigned to user	Follow On Task
Outstanding Architecture Compliance Tasks	All follow-on tasks in the Pending, Open, or Work in Progress state	Follow On Task

Report	Description	Table
Upcoming Architecture Compliance Audits	All scheduled audits	Audit

Architecture Compliance Module



Desired State

Desired State performs scheduled or on-demand audits of CMDB data to determine which records match the expected attributes, CI relationships, and relationships to other records in the system.

For example, desired state can determine if a computer has a license for a particular software program. The compliance process checks configuration items (CI) to ensure that their attributes and relationships comply with standards set by your organization. Audit

results show any discrepancies in the desired state of a record, and ServiceNow automatically assigns follow-on tasks to qualified users who can remediate those discrepancies.

Desired State roles

To access or configure certification elements, a user must have the certification_admin role. These users can create, update, and delete filters if they have the proper access to necessary tables.

In the base system, certification_admin users have limited system rights and do not have access to all the necessary tables. When assigning compliance resources, make sure to grant additional roles to the certification_admin user as needed. For example, the certification administrator requires roles that grant access to these tables:

- Company [core_company]
- Cost Center [cmn_cost_center]
- Schedule [cmn_schedule]

Desired State process

The desired state certification process can mean checking servers to ensure that their physical resources, such as CPU speed or memory, comply with certain standards. This process also ensures that all critical business services have a manager, support group, and approval group assigned.

The administrator responsible for certification creates definitions of desired states and then schedules an audit to check CIs for compliance. The audit results identify CIs that pass certification and itemize the discrepancies in those CIs that fail. The ServiceNow system automatically generates follow-on tasks to track the process of adjusting the CIs to the desired state.

Desired state differs substantially from data certification. Data certification is a manual process to ensure that your data matches reality. Desired state examines the same data and determines when the configuration of each item is in the desired and approved state.

1. Create a certification filter: Create a filter that defines a subset of configuration items to certify. You can create multiple versions of a

filter, and then activate the version you want to use for certification. You can create filters on the Configuration Item [cmdb_ci] table and all tables that extend it.

2. Create a template: Create a template with conditions that define the desired state of the physical attributes, related records, and relationships for a CI. The certification filter you select for the template determines which configuration items the system examines.
3. Create and run an audit: Create an audit using the template. Set the audit to run on a schedule or on demand. The audit generates a set of results based on the conditions from the template you specify. Determine usage of follow-on tasks:
 - Determine if the audit creates follow-on tasks and assignment.
 - Determine if the same follow-on task is used for the same audit failure across multiple runs. The system attribute glide.allow.new.cert_follow_on_task is set to true by default, allowing for new follow on tasks to be created for the same failure, at each audit run.
4. View audit results: View the audit results which display any discrepancies between the desired state, as specified by the template, and the actual state of the target configuration items.
5. Correct discrepancies: Correct the discrepancies the audit found by completing the follow-on tasks created by the system.

The Desired State Overview module displays various desired state reports. The Overview module is a type of homepage.

Desired State Overview module roles

Only compliance users with certain roles can access the Overview module. The different levels of access are:

Access levels per role

Role	Access
certification	View (view overview page and refresh reports)

Role	Access
certification_admin	<ul style="list-style-type: none">• View (view overview page and refresh reports)• Customize (refresh, add, delete, and rearrange reports) <p>View, customize</p>
admin	<ul style="list-style-type: none">• View (view overview page and refresh reports)• Customize (refresh, add, delete, and rearrange reports)• Edit (can edit reports)

The different levels of access are:

- View: can view the overview page and refresh reports.
- Customize: can refresh, add, delete, and rearrange reports.
- Edit: can edit reports.

Use the Desired State Overview module

The Desired State Overview module displays various desired state reports.

Before you begin

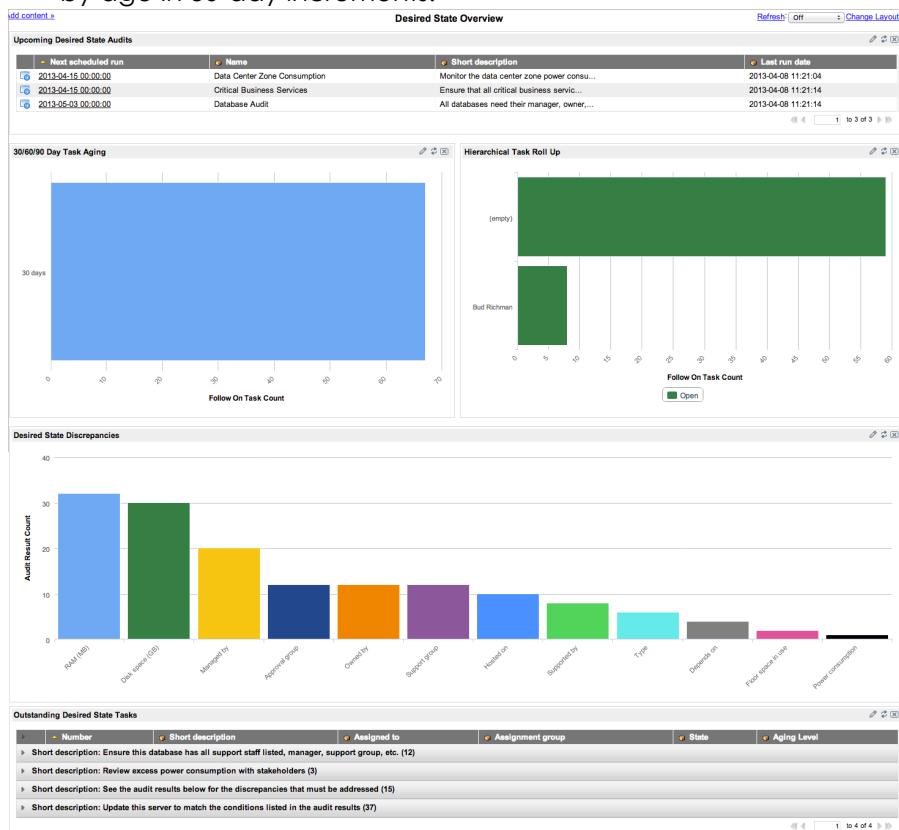
Role required: none

Procedure

1. Navigate to **All > Compliance > Desired State > Overview**.
2. Move or add reports where needed.
3. Click elements within the reports to obtain more information.

The Desired State Overview Module in the base system contains these reports:

- Upcoming Desired State Audits: All scheduled audits.
- Outstanding Desired State Tasks: All follow-on tasks in the **Pending**, **Open**, or **Work in Progress** state.
- Hierarchical Task Roll Up: All follow-on tasks grouped by **Assigned to** user.
- Desired State Discrepancies: All audit discrepancies for attributes and relationships.
- 30/60/90 Day Task Aging: All outstanding follow-on tasks grouped by age in 30-day increments.



The Desired State application includes reports to assess your audit results.

These reports are available to all users whose role gives them access to the [Reporting](#) application. Users with the admin role can share these reports with specific users or groups or change the display options.

Navigate to **Reports > View / Run**. In the Reports search field, enter all or part of the report name such as **Desired State**. You can also scroll to the designated category and select one of the reports.

In addition to these reports, you can generate your own reports.

Desired state report table

Report	Description	Category
Desired State Discrepancies	<p>This report displays all desired state audit results that have a follow-on task that is not yet in the Closed Complete state. This report displays by column name.</p> <ul style="list-style-type: none">• Type: bar chart• Table: Audit Result [cert_audit_result]	Audit Result
Desired State Result with Stability Unstable	<p>This report displays all audit results where the Stability field has the value Unstable. This report displays by CI and stacked by audit.</p> <ul style="list-style-type: none">• Type: bar chart• Table: Audit Result [cert_audit_result]	Audit Result

Report	Description	Category
Desired State Result with Threshold Exceeded	<p>This report displays all audit results where the Threshold field has the value Exceeded. This report displays by CI and stacks by each audit.</p> <ul style="list-style-type: none"> • Type: bar chart • Table: Audit Result [cert_audit_result] 	Audit Result
Upcoming Desired State Audits	<p>This report displays the desired state audits that are scheduled to run in the next two quarters.</p> <ul style="list-style-type: none"> • Type: List (tabular) report • Table: Audit [cert_audit] 	Audit
30/60/90 Day Desired State Task Aging	<p>This report displays the number of follow-on tasks that are not Closed Complete for desired state audit types. The report is grouped by aging level.</p> <ul style="list-style-type: none"> • Type: Horizontal bar chart 	Follow On Task

Report	Description	Category
	<ul style="list-style-type: none"> Table: Follow On Task [cert_follow_on_task] 	
Desired State Hierarchical Task Roll Up	<p>This report displays similar data to the Task Aging report, but groups the results by manager.</p> <ul style="list-style-type: none"> Type: Horizontal bar chart Table: Follow On Task [cert_follow_on_task] 	Follow On Task
Outstanding Desired State Tasks	<p>This report displays similar data to Task Aging report, but groups the results by short description.</p> <ul style="list-style-type: none"> Type: List (tabular) report Table: Follow On Task [cert_follow_on_task] 	Follow On Task

Certification audits

A certification audit compares the actual attributes of certain ServiceNow records. This audit selects a filter, against the expected attributes, relationships, and related record values defined by template conditions or a script.

You can configure the audit to create and assign follow-on tasks to remediate any discrepancies the audit finds. Audit records use a standard ServiceNow scheduler to determine when to run. After an audit

runs, the results and follow-on tasks appear in related lists in the audit record.

Users with the certification_admin role can create, update, delete, and run audits. Users with the certification role can view audits, audit results, and follow-on tasks.

Create a compliance audit. Compliance offers two types of audits: one uses templates to define conditions and the other uses a script.

Before you begin

Role required: certification_admin

Procedure

1. Ensure that an appropriate template record was created for this audit.

Note: Conditions in the template define the values to audit.

2. Use the CI Class Manager:

- a. Navigate to **All > Configuration > CI Class Manager**.
- b. Select a class from the **Class Hierarchy**.
- c. In the sidebar on the right, check **Advanced** and then click **Audit** in the **Compliance** group.

3. Or, navigate to one of these modules:

- **All > Compliance > Audits**
- **All > Compliance > Architecture Compliance > Audits**
- **All > Compliance > Desired State > Audits**
- **All > Compliance > Scripted Audits > Audits**

4. Click **New**.

The system opens a new record for the audit type associated with the navigation path you selected. The **Audit type** field is read-only.

5. Complete the form using the fields described in the table below.

6. Right-click the header bar and select **Save**.

The **Audit Results** and **Follow On Tasks** related lists appear on the form.

7. To run the audit immediately, click **Run Audit**.

When template audits run, ServiceNow updates the date and time in the **Last run date** field and populates the related lists. For scripted audits, the **Last run date** field is not populated.

8. View the records that passed and the discrepancies found by the audit in the **Audit Results** related list.

You can open template records and any follow-on tasks directly from this related list. Notice that the value in the **Task description** field appears as the **Short description** in the follow-on tasks.

Note: You cannot delete audit records that have audit results or audit results that have follow-on tasks. ServiceNow disables the **Delete** option in records and lists where these dependent records exist.

Creating Audits

Field	Description
Name	Name for this audit.
Filter	Filter to use when the audit type is Scripted. This field is required for scripted audits, but is hidden for all other audit types.
Template	[Required] Template to use when this audit runs. Audit type filters the list of available templates, and only the active versions of templates are available for selection. For example, when you create an audit from Desired State, only templates of the Desired State audit type are available for selection. For the Desired State

Field	Description
	and Architecture Compliance audit types, only templates for tables that extend the Configuration Item [cmdb_ci] table are available. This field is hidden when the audit type is Scripted .
Table	[Read-only] Table for the template.
Create tasks	Option to create follow-on tasks for correcting discrepancies (selected). In a scripted audit, you can create the logic for either task state by using true to create tasks or false to not create tasks. By default, this check box is cleared (false) in a new audit record.
Assignment type	<p>Method for assigning follow-on tasks. This field is visible only when the Create task check box is selected. Choices are:</p> <ul style="list-style-type: none"> • User Field: Select a user reference field on the table being audited. For example, you choose the user identified in the Managed by field on the failed record to perform the tasks. This selection displays the Assigned to and Assign to empty fields. If the reference field on the record is empty, the value in the Assign to empty field is used. • Specific User: Select a specific user to perform the

Field	Description
	<p>tasks. This selection displays the User field.</p> <ul style="list-style-type: none"> • Group Field: Select a group reference field on the table being audited. For example, you choose the group identified in the Support group field on the failed record to perform the tasks. Tasks are assigned to all members of the group. This selection displays the Assign to group and Assign to empty fields. If the reference field on the record is empty, the value in the Assign to empty field is used. • Specific Group: Select a specific group to perform the tasks. This selection displays the Group field. All members of the selected group are assigned to the tasks.
User	<p>The specific user this audit assigns to follow-on tasks. This user must have the certification role. This field is available under these conditions:</p> <ul style="list-style-type: none"> • Assignment type is set to Specific User. • Assign to empty is set to Create Assigned Task, and Assignment type is set to User Field.

Field	Description
Assign to group	<p>The group field that defines which group this audit assigns to the follow-on task. This field is available only when the Assignment type is Group Field.</p>
Group	<p>The specific group this audit assigns to follow-on tasks. This field is available only when the Assignment type is Specific Group.</p>
Assign to	<p>The user field that defines which user this audit assigns to the follow-on task. This field is available only when the Assignment type is User Field.</p>
Assign to empty	<p>The behavior to use if the field selected in Assign to or Assign to group is blank on the record being audited. For example, if a follow-on task must be assigned to a manager, but no manager is identified, the Assign to empty setting determines what happens. This field appears only when the Assignment type is User Field or Group Field. Choices are:</p> <ul style="list-style-type: none"> • Do Not Create Task: No follow-on task is created when the Assign to or Assign to group field is empty. • Create Unassigned Task: Create a follow-on task, but do not assign it to any user or group. The task can be manually assigned later.

Field	Description
	<ul style="list-style-type: none"> • Create Assigned Task: Create a follow-on task and assign it to the user or group specified. If the assignment type is User Field, the User field becomes available. If the assignment type is Group Field, the Group field becomes available. <p>The audit automatically creates follow-on tasks for all records that have Assign to populated, regardless of the Assign to empty setting.</p>
Short description	Brief description of the purpose of the audit.
Task description	General description of the work required for the follow-on tasks for the audit. All follow-on tasks created by this audit inherit this description.
Active	Activation control for this audit record. Clear this check box to prevent this audit from running and creating follow-on tasks.
Run	<p>How often to run the schedule that generates the audit.</p> <ul style="list-style-type: none"> • Daily • Weekly • Monthly • Periodically • Once

Field	Description
	<ul style="list-style-type: none"> On Demand
Day	<ul style="list-style-type: none"> If Run is Weekly, the day of the week when the audit runs. If Run is Monthly, the day of the month when the audit runs. If the day is 29, 30 or 31, for shorter months the audit runs on the last day of the month.
Repeat Interval	If Run is Periodically , the frequency that the audit runs, based on a 24-hr. clock. Enter the number of days between audits and the time of day that you want the audit to run. For example, set Days to 10 and Hours to 14:00:00 to run the audit every 10 days at 2:00pm.
Starting	If Run is Periodically or Once , the date and time when the audit runs.
Time	If Run is Daily , Weekly , Monthly , or Once , the time of day, on a 24-hour clock, when the audit runs.
Last run date	[Read-only] The last date and time the audit ran, either on its regular schedule or manually. Audit previews do not update this field.
Next scheduled run	[Read-only] The next date and time when the audit runs. The

Field	Description
	system recalculates this field when you change the schedule.
Audit type	<p>[Read-only] The type assigned to this audit. The system selects the audit type based on the application from which the audit is created. The type can be:</p> <ul style="list-style-type: none"> • Desired State • Architecture Compliance • Compliance • Scripted
Health window	Duration of the evaluation period for threshold and stability. The health window value defines the number of Health window units in an evaluation period for an audit. This value is expressed as a positive integer. The default value for this field is 7 .
Health window unit	Unit of measurement that defines the duration of a health window. The default value for this field is Days . Choices are: <ul style="list-style-type: none"> • Minutes • Hours • Days • Months
Threshold count	Sets the acceptable number of audit failures for the desired state field that can occur within the specified health window for a CI.

Field	Description
	The audit results indicate when a desired state field is within or has exceeded this threshold limit. The default value for the threshold is 5.
Stability count	Sets the acceptable number of times that audit results for a CI can switch between Certified and Failed within the specified health window. The audit results for a CI indicate whether it is stable or unstable. The default value for stability is 1.
Run this script	Audit script to run which contains the conditions that a CI need to comply with to pass the audit. This field is available only when the audit type is Scripted . The Audit form includes a sample script with instructions for performing the audit and generating the follow-on tasks.

New audits can be created from an existing audit.

1. Open the audit record you want to copy.
2. Change the name or short description to distinguish this audit from the original.
3. Make any other changes you need.
4. Right-click in the header bar and select either Insert or Insert and Stay from the context menu.
The system clears the **Last run date** field and inserts the record into the database.

The system performs audits automatically from the schedule you configure.

Users with the certification_admin or admin role can generate on-demand audits directly from the Audit form by clicking **Run Audit**. When an audit runs, ServiceNow populates the **Audit Results** related list in the form and shows follow-on tasks, if any, in the **Follow On Tasks** related list. Click **Preview Audit Results** to generate an audit preview that tests your template conditions without generating any audit results.

Audit results show the records that have passed or failed an audit and itemize any discrepancies detected.

A discrepancy is considered any departure from the expected conditions defined in the template or script used for the audit. Audit results provide links to the source records and to the follow-on tasks for bringing failed records into compliance. Records that pass an audit have a single entry in the results table with a state of Certified. Records that fail an audit show all discrepancies, each with a state of Failed.

ServiceNow displays results from a certification audit in these locations:

- Audit Results list
- A related list in the Audit record
- A related list in the compliance view of a CI record

To generate certification results, you must first create and run an audit.

Before you begin

Role required: none

Procedure

1. Navigate to one of the following locations:
 - **Compliance > Desired State > Audit Results**
 - **Compliance > Architecture Compliance > Audit Results**
 - **Compliance > Scripted Audits > Audit Results**
 - **Data Certification > Schedules > Audit results**
2. From any audit results list, you can edit the filter to show the results for any audit type.

Audit Type filter



The results filter by audit type and grouped by audit number. Within the groups, results list by date, from oldest to newest.

3. You can open the audit record, the CI record, or the follow-on tasks from this list.

Note: The **Audit type** field was set automatically when the audit result was created and cannot be changed. For scripted audits, the audit type is set when you create the audit record.

Audit results show this information:

Audit Results

Field	Description
Created	Date and time the audit ran.
Document	Record that was certified, such as a configuration item (CI).
State	<p>Results of certification for each condition evaluated. The three possible states are:</p> <ul style="list-style-type: none"> • Certified A certified record is one that passed all conditions. ServiceNow generates only one audit result for a certified record. • Failed Records that are not certified have an audit result for each failed condition. The Column name, Desired value, Discrepancy value, and Follow on task are only populated for failed results.

Field	Description
	<ul style="list-style-type: none"> Pending A pending state indicates that the audit is incomplete. Data certification audits use this state when a result is awaiting user input.
Column name	Audited field, relationship, or related list column that did not match the expected state.
Desired value	Attribute or relationship required for this record that was not found, from the condition in the expected state template. For data certification, this column is blank if the record has a state of Failed or Pending.
Discrepancy value	Actual value of the attribute that did not match the expected state. The follow-on task, if provided, tracks resolution of this discrepancy. In a list of results for the Data Certification audit type, this column is blank if the record has a state of Certified or Pending.
Follow on task	Link to the follow-on task generated for remediating a discrepancy.
Audit	Link to the audit record that produced the results.
Threshold	State of an audited, desired state field with a defined failure threshold. This threshold is the acceptable number of failures for a desired state field within

Field	Description
	a specified health window and is configured in the Audit form. Possible threshold states for the results are: <ul style="list-style-type: none">• In Limit• Exceeded
Stability	Stability state of a CI. Stability state is based on the number of times the audit result for a desired state field changes from Certified to Failed within a specified health window. Possible stability states are: <ul style="list-style-type: none">• Stable• Unstable

While audit results can be deleted, you cannot delete an audit result that has a follow-on task associated with it.

Before you begin

Role required: none

Procedure

1. Navigate to one of these modules:
 - **Compliance > Desired State > Audit Results**
 - **Compliance > Architecture Compliance > Audit Results**
 - **Compliance > Scripted Audits > Audit Results**The list groups by audit name.
2. Select the checkbox for a result in the list, and then select **Delete** from the **Actions on selected rows** menu at the bottom of the list.

Note: If the result record has a follow-on task, the **Delete** option is not available. If you select multiple records, some with and some without tasks, the system only deletes those records that do not have tasks.

3. Click a date/time link to see the results for a specific CI.

Note: The **Delete** button only appears on the form if the audit result does not have a follow-on task.

4. Click **Delete**.

After an audit has run, you can view the results and follow-on tasks from the Compliance view in the records of every CI audited.

Before you begin

Role required: none

About this task

This view is available only for systems that use the default CI classes provided with the base ServiceNow system, such as Hardware, Software, and Computer. For information about creating views, see View Management.

Procedure

1. Navigate to **All > Configuration** and open the record of a CI that was included in a compliance audit.
2. Select the view to configure by performing the appropriate action for your list version.

Version	Action
List V2	Open the context menu and select View > Compliance .
List v3	Open the context menu and select Change View , and then click Compliance .

The Audit Results Compliance View appears.

Audit Results Compliance View List Descriptions

Lists	Description
Passed Audit Results	Lists audits for this CI that passed without discrepancies. The information includes the versions of the template and filter used. Records are grouped first by audit, and then by creation date and time.
Failed Audit Results	Lists all failed audits for this CI. The information includes the discrepancy data, the follow-on task, and the versions of the template and filter used. Records are grouped first by audit, and then by creation date and time.
Follow On Tasks	Lists all follow-on tasks generated from audit discrepancies for this CI.

3. Right-click the header bar and select **View > Compliance** from the context menu.

You can preview an audit to view potential results without saving audit results or generating follow-on tasks. For example, use this feature to test template conditions for correctness without creating thousands of result records.

In an audit record, click **Preview Audit Results** under **Related Links** to show a summary of the potential audit results appears at the top of the audit record.

Previewing does not change the **Last run date** field.

A health window is a trailing time frame in which the ServiceNow system evaluates audit results from CIs that have desired state fields defined.

The **Health window** and **Health unit** fields define each window, and ends when an audit runs. For example, an audit runs on the fifteenth of the month with a seven-day window. It evaluates the threshold values of a desired state field from the eighth to the fifteenth. When the same audit runs the next day, the system evaluates the threshold from the ninth to the 16th, and so on. The audit counts backward seven days from the current day. ServiceNow evaluates a CI threshold value for each health window, without considering the results from the previous window. As a result, the health of a CI can fail for one audit and then pass in a subsequent audit that runs in a new window.

ServiceNow evaluates stability by recording the number of times a desired state threshold value for a CI switch between **Failed** and **Certified** within the health window. In the example shown here, a 5-minute health window was set for the desired state field on a UPS unit that measures the remaining battery time. The threshold was set at 2, which allows the field to fail two audits in the same health window.

In the initial audit, the system evaluated the threshold value for the **Seconds on battery** field within a 5-minute window. This window ran from 13:52:51 to the time of the audit at 13:57:51. The desired state field showed **In Limit** for that audit and the second audit conducted less than a minute later. The next two audits were conducted within five minutes of the first audit and both showed that the threshold (set at 2) was **Exceeded**. A subsequent audit was conducted five minutes after the audit in which the desired state field threshold was first exceeded. Since the health window had moved forward enough units, the **Seconds on battery** field was within limits again with only one failure in the 5-minute window being evaluated.

Certification filters

A certification filter creates a subset of ServiceNow records to audit, typically from configuration items (CI) of a certain type, such as all UNIX servers in a specific datacenter.

However, you can define a filter for any ServiceNow table by using any set of system-supported conditions. Audited records identified by a filter for expected attributes or relationships, depending on the audit type.

You can create multiple versions of a filter, reactivate inactive versions, and select the version you want to use in a template or a certification schedule. Only the active versions of a filter are available for selection

in template records. You can use a single filter for multiple certification templates or schedules.

Certification filters

Filter	Description
Data Certification	Validates CMDB data.
Architecture Compliance	Manages reviews of CMDB data in architecture compliance audits to determine which configuration items (CIs) match expected attributes.
Desired State	Manages reviews of CMDB data to determine which CIs match a desired state for both attributes and relationships.
Compliance	Manages reviews of records from any ServiceNow table to determine which records match an expected set of attributes and related record conditions.
IT Governance Risk and Compliance	Generates audits and tests to ensure that controls are being followed and creates tasks to track corrective actions.

Roles

In the base ServiceNow system, users with the certification_admin role have limited system rights and do not have access to the tables required for creating a filter.

When assigning compliance resources, make sure certification_admin users have any additional roles they need. For example, a user requires roles that grant access to the Company [core_company] table.

The compliance filter for license bases uses the following fields to define entitled users or CIs.

These field values can be used independently or together to calculate compliance.

Compliance Filter

Value	Description
Entitled company	All users or configuration items (CI) at all locations of this company, in all departments are entitled to use this software package. Compliance at this level calculates how many licenses are purchased for the company at large and how many entitled users or CIs consume them.
Entitled location	CIs and users who are assigned to this company location in any department are entitled to use this software package. Compliance at this level calculates how many licenses are purchased for this company location and how many users and CIs consume them.
Entitled department	Only the users or CIs in this department at this company location are entitled to use this software package. Compliance is calculated for a single department only.

The license form can display information about all CIs or named users who are using this software package. The form indicates when license reconciliation is necessary and displays all compliant users or CIs.

Possible compliance levels are:

Compliance

Level	Description
Non applicable	Compliance levels for all infrastructure licenses that are related to cluster licenses are set to Non applicable automatically. Compliance levels are calculated in the cluster license only, and not in the related infrastructure licenses.
Out of compliance	More licenses are being consumed than were purchased. There are more users or Cls using this license than the license allows, and some users or Cls are not be entitled to use this software package.
Unused	The licenses for this software package are currently unused.
Reconciliation required	Clis or users who are not entitled to use this software are consuming licenses. Licenses that require reconciliation are considered out of compliance. Reconciliation requires action to ensure that unentitled users are not using the software. Reconciliation involves uninstalling software or increasing license counts to match actual user counts.
Nearly out of compliance	For a software package to be at this compliance level, more than 95% of the licenses are in use by entitled users or Cls. License bases at this level are considered to be In compliance .

Level	Description
In compliance	This software package has unused licenses. All users or CIs using a license are entitled to use this software package.

You can create as many versions of a filter as necessary. You can then designate which versions are active and available for selection in Compliance template records, Governance Risk and Compliance control test definitions, or Data Certification schedule definitions.

1. Navigate to **All > Configuration > CI Class Manager**, and:
 - a. Click **Hierarchy** to display the CI Classes list.
Select the class to create a filter for.
 - b. In the class navigation bar, expand **Health**, select **Compliance**, and then click **Certification Filter**.
2. Or, navigate to one of these modules:
 - **Compliance > Filters**
 - **IT GRC > Administration > Filters**
 - **Data Certification > Schedules > Certification Filters**
3. Select an existing filter to edit, or click **New**.
4. Fill in the fields (see table below).
5. Click **Submit**.
This action saves the filter as version 1.
6. To create another version of this filter, open the record and modify the name, table, or conditions.

Note: You can change a filter Description without incrementing a version.
7. Click **Update**.

The system saves a new version of the current filter and makes it the Active version. The previous version is marked inactive. The system displays only active filter versions for selection when you create templates or schedules.

Creating Filters

Field	Description
Number	[Read-only] Displays the automatically assigned filter identification number. All versions of a filter have the same number.
Name	[Required] Filter name.
Description	[Optional] Describes this filter. You can change the description of a filter without incrementing a version.
Table	Specifies the table containing the records to select. The template or schedule that uses this filter works on this table. For example, select the ESXi Server [<code>cmdb_ci_esx_server</code>] table to select VMware ESX servers.
Active	Makes this filter available for use from the Filter field on the Certification Template or Schedule Definition form. Multiple versions of a filter can be active. You can activate or deactivate a filter without incrementing the version.
Version	[Read-only] Indicates the version of this filter. Any changes to this filter, except to the description or the Active check box, makes it

Field	Description
	inactive. The system increments the version of the updated filter and marks it as active. The system saves all versions of the filter and makes them available for reactivation.
Filter condition	Specifies the fields, operators, and values that create the filter. The available fields are based on the table selected. The condition builder shows the number of records that match the conditions. Click the refresh icon Refresh Conditions  to recalculate the number of matching records when you edit the conditions.

New filters can be created from an existing filter.

1. Navigate to **All > Configuration > CI Class Manager**, and:
 - a. Click **Hierarchy** to display the CI Classes list.
Select the class to create a filter for.
 - b. In the class navigation bar, expand **Health** and select **Compliance**.
Then click **Certification Filter**.
2. Or, navigate to one of these modules:
 - **Compliance > Filters**
 - **IT GRC > Administration > Filters**
 - **Data Certification > Schedules > Certification Filters**
3. Open the filter record that you want to copy.

4. Make sure to change the filter name or description to distinguish the new filter from the original.
5. Make any other necessary changes.
6. Right-click in the header bar and select either **Insert** or **Insert and Stay** from the context menu.

The system increments the record number and sets the version to 1 for the new record. Both the original filter and the copy are Active and appear in the record list. Showing all copies of a filter allows you to see the entire history of the filter.

Only users with the certification_admin or admin role can delete filter versions. But, you cannot delete a filter that is being used in a template or a scripted audit.

Before you begin

Role required: none

About this task

You cannot delete a filter that is being used in a template or a scripted audit.

Procedure

1. Navigate to **All > Configuration > CI Class Manager**, and:
 - a. Select **Hierarchy** to show the CI Classes list. Select the class to delete a filter for.
 - b. Expand **Health** in the class navigation bar, and select **Compliance**. Then select **Certification Filter**.
2. Or, navigate to one of these modules:
 - **Compliance > Filters**
 - **IT GRC > Administration > Filters**
 - **Data Certification > Schedules > Certification Filters**
3. Open the filter record you want to delete.

- a. To delete a single filter version, open that version record and click **Delete**.
The system hides the **Delete** button for filters that are in use. If you delete the latest version of a filter that is active, the previous version of that filter is reset to Active.
 - b. To delete all unused and inactive versions of a filter, open any version of that filter and click **Delete inactive versions** under **Related Links**.
4. When prompted, select **OK** to proceed.
The system deletes unused filter versions. A message in the header bar identifies filter versions that cannot be deleted because they are used in a template or scripted audit.

You can view and manage all versions of a filter from the Certification Filter form.

Before you begin

Role required: none

About this task

Versions can be displayed in a list. The default list of filters displays only the active version of each filter. To see all filter versions in the list view, select **All** in the breadcrumbs.

Procedure

1. Open any version of a filter.
The Other Versions related list displays all other versions of this filter, both active and inactive. The system prevents you from editing either the filter version or the record number in the list view.
2. Click any version in the related list to display the record for that version.
3. To make an inactive filter the current version, open the filter, edit it if desired, and then click **Revert**.
This action:
 - Deactivates the previous active version of the filter.

- Copies the inactive filter.
- Makes this new copy current and active.

Certification follow-on tasks

The ServiceNow system can automatically generate and assign follow-on tasks to correct discrepancies detected during compliance audits.

The system attribute glide.allow.new.cert_follow_on_task is set to true by default, allowing for new follow on tasks to be created for the same failure, at each audit run. You can set this property to false, to configure audit to use the same follow-on task for the same audit failure across multiple runs.

You configure and assign follow-on tasks to qualified users or groups in the audit record. A user with the certification_admin role can reassign any follow-on task. The Audit Results related list in the Follow On Task form contains links to the records that failed.

Access follow-on tasks

Users with the certification role can only access follow-on tasks assigned to them but can reassign these tasks to other users.

1. Navigate to **All > Compliance > My Follow On Tasks**.
The list contains all active follow-on tasks assigned to the logged in user.
2. Open a task.

The record shows the specifics of the task, the task activity, and the failed audit results.

3. Open records from the **Audit Results** related list to see each discrepancy.
4. Go to the CI named in the record and perform the work to bring it into compliance.
5. Update the **State** field in the follow-on task record and add work notes as you correct each discrepancy.

When you change the state, the system updates the task activity appropriately.

When the task is **Closed Complete** it no longer appears on the **My Work** list.

Manage follow-on tasks

Users with the certification_admin or admin role can see all follow-on tasks.

Before you begin

Role required: none

About this task

Tasks are pre-assigned to a user or group as specified in the audit record, but users with the certification_admin role can reassign the task.

Procedure

1. Navigate to the appropriate application:

- **Compliance > Architecture Compliance > Follow On Tasks**
- **Compliance > Desired State > Follow On Tasks**
- **Compliance > Scripted Audits > Follow On Tasks**

The list of follow-on tasks appears, filtered by audit type.

2. Open a task.

The **Audit** and **Configuration item** fields are read-only for all users.

3. Edit the **Change group** or the **Assigned to** field if necessary.

4. Edit the **Short description** field if necessary.

The short description is inherited from the **Task description** field in the Audit form.

5. Use the links in the **Audit Results** related list to open the individual records that failed the audit.
6. If you update the follow-on task record, be sure to add work notes.

Certification templates

Certification templates can define attributes, relationships, and reference field values that indicate what a record is expected to contain.

These values are used to perform audits on ServiceNow records. The certification filter selected in the template identifies the table and records to audit, and the template conditions set the expected state for those records. The type of audit you create determines which tables and template conditions are available.

Users with the certification_admin role can create, update, and delete templates. Users with the certification role can view template versions.

Certification template audit types

When you create a template, ServiceNow assigns an Audit type that determines which tables and conditions are available in the certification template. This value is based on the application from which the template is created. Each application lists only the templates with the associated type.

Available Condition Builders

The available condition builders for each audit type:

- Compliance: Runs audits on any set of ServiceNow records, not only configuration items (CI). This audit type provides the following types of conditions for any ServiceNow table:
 - Attribute: Sets conditions for the attributes of the records.
 - Related List: Runs audits on records in tables that reference the table defined in the template.
- Architecture Compliance: Defines the following types of conditions for tables that extend the Configuration Item [cmdb_ci] table.

- Attribute: Sets conditions for physical attributes of CIs, such as memory or disk size.
- Related List: Runs audits on records in tables that reference the table defined in the template.
- Desired State: Defines the following types of conditions for tables that extend the Configuration Item [cmdb_ci] table.
 - Attribute: Sets conditions for physical attributes of CIs, such as memory or disk size.
 - CI relationship: Defines the relationships these CIs have with other CIs. An example of a relationship is a business service, such as Outlook Web Access, that depends on a server.
 - User relationship: Defines the user who reviewed the log records. The only operator available with this condition builder.
 - Group relationship: Defines user groups who backed up this CI. The only operator available with this condition builder.
 - Related List: Runs audits on records in tables that point toward the table defined in the template.

Certification Template Record List

The default Templates list displays only the active version of each template, but users can update the breadcrumbs to display all template versions.

- Default Templates List: The default Templates list displays only the active version of each template, filtered by **Audit type**.
- All Template Versions: To view all template versions for an audit type, click the arrow before **Active=true** to remove that condition from the breadcrumbs. 

To create a certification template, follow these instructions.

Before you begin

Activate the Certification Core plugin to enable the Compliance functionality. See [Compliance Activation](#) for details.

Procedure

1. Ensure that you have an appropriate filter that defines the records the template evaluates.

The template applies its conditions to these records.

2. Use the CI Class Manager to navigate to the Certification Template form:

- a. Navigate to **All > Configuration > CI Class Manager**.
- b. Click **Hierarchy** to display the CI Classes list.
Select the class for which to create a certification template.
- c. In the class navigation bar, expand **Health** and then click **Compliance**.

3. Click **Certification Template**.

4. Or, navigate using one of these paths:

- **All > Compliance > Architecture Compliance > Templates**
- **All > Compliance > Desired State > Templates**
- **All > Compliance > Templates**
- **All > Audit Definitions > Templates**

5. Click **New** or select a certification template to edit.

The following fields are completed automatically:

- **Number:** Each new template has a unique number. All versions of the same template use the same number.
- **Active:** All new templates are set to Active.
- **Version:** The version of a new template is set to 1.
- **Audit type:** The system sets the default type to Architecture Compliance, Desired State, or Compliance, depending on the application in which the template was created. You can select a different type when you create the template, but the field becomes read-only when you submit the record. The system

uses audit types to filter record lists for appropriate data and determine which conditions are visible on the template form.

6. Complete the following mandatory fields.

- **Name:** Enter a descriptive name for this template. The name helps identify the purpose.
- **Filter:** Select the filter that identifies the records to be certified. You can select either active or inactive filter versions. By default, the system presents only active versions for selection. If you start typing the name of a filter, the auto-complete feature displays all versions for selection. For architecture compliance and desired state templates, only filters that use a table extended from Configuration Item [cmdb_ci] appear on the choice list. All filters appear on the choice list for a compliance template. After you select a filter, the template condition builder appears. The template operates on the table specified in the filter.

7. Enter a **Description** for this template.

8. Define certification conditions using the condition builders.

All conditions are AND conditions. The audit type of the template determines which conditions are available.



Certification Attribute Conditions: [All audit types] Select configuration item attributes or specifications to certify, such as CPU count, memory, or disk space. Available fields in the attribute condition builder depend on the table from the filter. Typical ServiceNow conditions for attributes are available, including the between operator for setting numerical conditions with high and low boundary values. This operator was added specifically for desired state conditions.

The **Show Related Fields** item supports dot-walking, allowing you to include referenced fields in a certification attribute condition. Click **Show Related Fields** or **Remove Related Fields** to add or remove referenced fields (in the form of <field> => <field>). Select a referenced field to drill down to the next level of referenced fields.

See [Dot walking](#).

- **Certification CI Relationship Conditions:** [Desired State audit types] Define the CI relationships to certify, such as Runs on or Depends on.
- **Certification User Relationship Conditions:** [Desired State audit types] Select the desired user relationship for this configuration item. The relationship provided in the base system is Log reviewed by.
- **Certification Group Relationship Conditions:** [Desired State audit types] Select the desired group relationship for this configuration item. The relationship provided in the base system is Backed up by.
-

Certification Related List Conditions: [All audit types] Select field values from tables that reference the template table, or user-defined related lists which are created via custom relationships in the sys_relationship table. To create a condition that evaluates all servers in the Server [cmdb_ci_server] table for the presence of Microsoft Word 2007, as referenced in the Software Installation [cmdb_sam_sw_install] table. The resulting condition is [Software Installation->Installed on] [Display name] [is] [Microsoft Word 2007].

Check **All** to include all records in the condition requirements of the related list. If there are no records in the related list, then:

- If **All** is checked, the condition requirement is met.
- If **All** is unchecked, the requirement is not met.

Note: By default, the condition builders for relationships display only suggested relationships. To see all possible relationships, select the **Show all relationships** check box on the right side of the form.

- a. Click **Insert a new row** to insert a condition.
You cannot insert an empty condition.
- b. Click the green check mark icon to save a condition.
Make sure to save the condition before performing any other operation. Updating the form does not save the condition.

- c. To delete a condition, click the red **X** beside the condition.
The system marks the condition as inactive.
- d. To reactivate a condition, click the gray **X**.
If another condition for the same field exists, the system prevents reactivation and warns you of the conflict.

9. Click **Submit**.

ServiceNow saves the template as version 1.

10. To create another version of this template, change the name, edit the conditions, or select a different filter.

Updating the template **Description** does not create a new version.

Note: If you select a filter whose table is incompatible with the existing template conditions, the system displays a warning that the conditions cannot be applied.

 Some template conditions are incompatible with the selected filter. Incompatible conditions will not be used for auditing.

11. Click **Update**.

The system saves a new version of the current template and makes it the Active version. The previous version is marked inactive.

New templates can be cloned from an existing template.

1. Open the template record to be copied.

2. Make any necessary changes.

3. Change the template name or description to distinguish it from the original.

4. Click **Clone**.

ServiceNow increments the record number above the highest template number and sets the version of the new record to 1. A message appears under the header bar naming the source record for the clone. Both templates are Active and appear in the record list. The record list allows you to see the entire history of the template.

You can view and manage all versions of a template from the Template form.

1. Open any version of a template.

The **Other Versions** related list displays all other versions of this template, both active and inactive.

2. Click any version in the related list to display the record for that version.
3. Update the template to create a new version.
The system increments a version of the template when you edit any field except **Description** and **Active**. You can manage the template versions without returning to the list view.
4. To make an inactive template the current version, open that version, edit it if desired, and then click **Revert**.
This action does:
 - Deactivates the previously active version of the template.
 - Copies the inactive template.
 - Makes the new copy the current, active version.
5. Select the **Audits** related list to view all audits configured to use this template.
6. Click **New** to create a new audit record with the template selection and table pre-populated.

Certification templates can be deleted.

About this task

Only users with the certification_admin or admin role can delete template versions. You cannot delete a template version that is being used for an audit.

Procedure

1. To delete a single template version, open that version record and click **Delete**.

The system hides the **Delete** button for templates that are in use. If you delete the latest, active version of a template, the previous version of that template is reset to Active.

2. To delete all unused and inactive versions of a template, open any version of that template and click **Delete inactive versions** under **Related Links**.

This control appears on all versions, whether they are used in an audit.

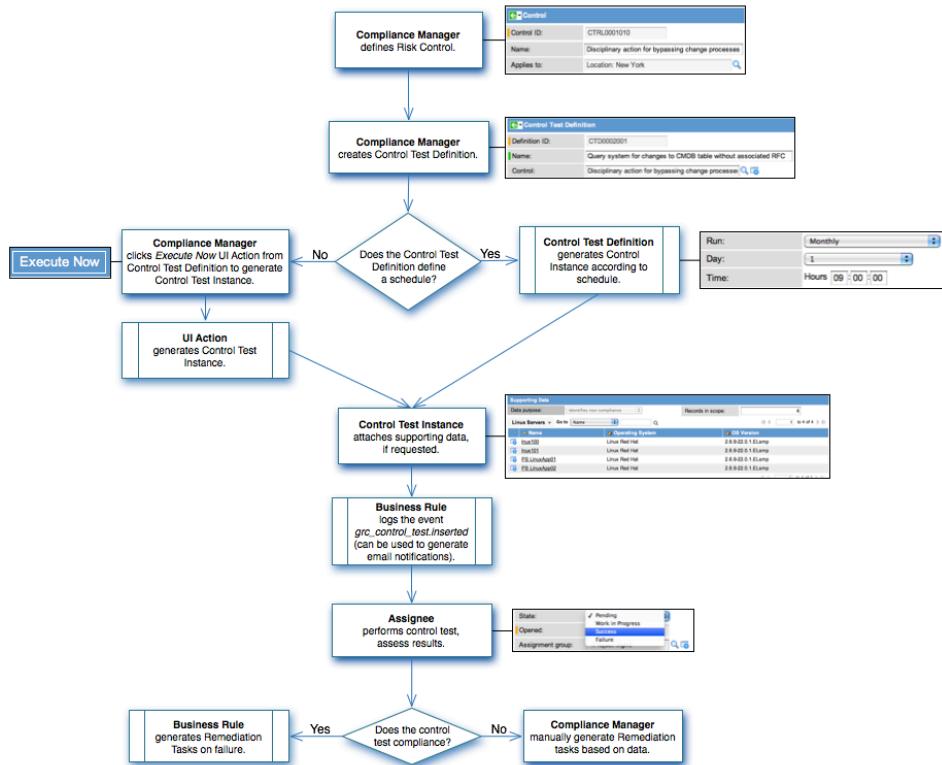
- When prompted, click **OK** to proceed.

The system deletes only template versions that are not used in an audit. All protected versions are named in a message that appears in the header bar.

Controls and tests management

After you identify the risks, define controls with accompanying control tests to prevent issues from occurring.

This diagram illustrates the entire IT GRC control process.
ITGRC Control Process



Define a control before you define a control test.

1. Navigate to **IT GRC > Controls > All**.
2. Click **New**.
3. Fill in the form, as appropriate (see table).
4. Click **Submit**.

Defining A Control

Field	Description
Control ID	A unique identifier generated dynamically by the system.
Name	A name for the control.
Applies to	Number of a record from any table in the system. This value defines the scope of the control.
Classification	The type of control.
Purpose	The approach that the control takes.
Control frequency	The basis for determining when the control is implemented.
State	A workflow field that determines where in the authoring process the control is.
Key control	Indicator that the control is considered key to preventing material risk, when selected.
Owning group	A reference to the group with ownership over the control.
Owner	A reference to the user with ownership over the control.

Field	Description
Owner delegate	A reference to the user who has ownership over the control when the specified owner is unavailable.
Description	A long-form description of the control.

After you define a control, create control tests that run periodically and provide documented evidence of whether the associated control is operating correctly.

1. Navigate to **All > IT GRC > Administration > Control Test Definitions**.
2. Click **New**.
3. Fill in the form, as appropriate (see table).
4. Click **Submit**.

Defining A Control Test

Field	Description
Definition ID	A unique identifier generated dynamically by the system.
Name	The name of the control test.
Control	A reference to the control being enforced.
Method	<p>One of the following choices for determining the test assignee:</p> <ul style="list-style-type: none">• Assign to Group: Assignment group for the control test.• Assign to Individual: User assigned to the control test.

Field	Description
Assign to group	Group assigned to this control test. This field is available only when the selected method is Assign to Group .
Assign to	User assigned to this control test. This field is available only when the selected method is Assign to Individual .
Remediation group	Group assigned to the remediation tasks when a control test fails.
State	A workflow field to indicate where in the drafting process this control test currently is. If the state is Active , control test instances are dynamically generated based on the record definition.
Run	Frequency for generating control test instances. Choices are: <ul style="list-style-type: none"> • Daily • Weekly • Monthly • Periodically • Once • On Demand
Time	The time that a control test instance is automatically generated when Run is set

Field	Description
	to Daily , Weekly , Monthly , or Periodically .
Day	Day of the week that a control test instance is generated each week when Run is set to Weekly . Day of the month if Run is set to Monthly .
Repeat interval	A duration, in days and hours, between the automatic generation of control test instances if Run is set to Periodically .
Starting	The date and time control test instances are first generated when Run is set to Periodically . The only date and time a control test instance is generated if Run is set to Once .
Execution step	The steps involved in the control test.
Expected result	The result that occurs after these tests.
Include supporting data	Indicator whether sample data is taken from a particular table within the instance when the control test instance is generated.
Data purpose	<p>The purpose of the data being sampled If Include supporting data is selected. This selection influences how the control test is performed. Choices are:</p> <ul style="list-style-type: none"> • None

Field	Description
	<ul style="list-style-type: none"> • Support test execution: Returns a random sampling of records. • Identifies non compliance: Returns all the records that do not match the condition or conditions specified. • Identifies compliance: Returns all the records that do match the condition or conditions specified.
Table	<p>The table from which to sample when Include supporting data is selected.</p> <p>This field is read-only when Template is the Condition type. When you select a template to define test conditions, the certification filter used in the template sets the table and cannot be changed.</p>
Fields	<p>The list of fields to pull values from when determining whether records match the conditions when Include supporting data is selected.</p>
Condition type	<p>The type of conditions applied to the table and fields. Choices are:</p> <ul style="list-style-type: none"> • Basic: Applies conditions to the table in question. • Advanced: Uses condition collections to apply

Field	Description
	<p>conditions to the table and to related tables.</p> <ul style="list-style-type: none"> • Template: Uses certification templates to apply conditions to the specified table. Select the template to use from the Template field.
Sample size	An integer number of rows for a random sample if Include supporting data is selected. A sample size of zero returns all matching records. This field is available only if Condition type is set to Basic and Data purpose is set to Support test execution .
Control test conditions	A condition builder that limits the sample data when Include supporting data is selected. This field is available only if Condition type is set to Basic .
In scope definition	A reference to a condition collection if Include supporting data is selected and Condition type is set to Advanced .
Configuration to retrieve	Method for using the Configuration reference field if Include supporting data is selected and Condition type is set to Advanced or Template . <ul style="list-style-type: none"> • None: Returns all records in scope. • Matching: Returns all matching records in scope.

Field	Description
	<ul style="list-style-type: none"> • Non-matching: Returns all non-matching records in scope. <p>For more information, see Defining Advanced Conditions.</p>
Template	<p>[Required] Certification template that defines conditions for this test definition. Only templates with an audit type of Compliance are available for selection. This field is available and mandatory when the value in the Condition type field is Template.</p>
Configuration	<p>Condition collection to use. This field is available only if Include supporting data is selected, Condition type is set to Advanced, and Configuration to retrieve is set to anything except None.</p>

Set the **Condition type** to **Advanced** on control tests to define more flexible conditions using condition collections.

Condition collections have one primary condition, which is applied to the selected table, and one or more supplemental conditions.

When a control test is performed, advanced conditions evaluate in this order:

1. The system processes the condition collection in the **In scope definition** reference in this order:
 - a. The primary condition is processed on the fields specified in **Table** and **Fields** on the control test definition, returning an array of elements.

- b. For each element in the array returned by the primary condition, supplemental conditions are processed, filtering the array of elements further.
 - c. The **In Scope** field is updated with the number of elements in the array.
2. The condition collection in the **Configuration** reference is processed on the array of elements returned from the **In scope definition**. The choices for **Configuration to retrieve** are:
 - **None**: These conditions are skipped. Supporting Data is all the elements that are in scope.
 - **Matching**: The control test checks the array of elements, returning any elements that match the **Configuration**.
 - **Non-matching**: The control test checks the array of elements, returning any elements where at least one condition did not match the **Configuration**.
 3. The final array of elements is recorded as **Supporting Data** records.

Both the **In Scope** and **Configuration** fields refer to the Condition Collection [grc_condition_collection] table.

To define condition collections:

1. Navigate to **IT GRC > Administration > Condition Collections**.
2. Click **New**.
3. Populate these fields:
 - **Name**: Name of the condition collection.
 - **Description**: Description of the condition collection.
 - **Type**: Which **Control Test Definition** field references the condition collection. Choices are:
 - **In Scope Definition**
 - **Configuration Definition**

4. After the condition collection is defined, use the **Add Condition** related link to add these conditions:

- **Condition:** Predefined condition definition from the Condition [grc_condition] table.
- **Condition type:** The condition collection **Type** determines the choices:
 - **In Scope Definition**
 - Primary
 - Supplemental
 - **Configuration Definition**
 - Not Applicable

To define new condition records:

1. Navigate to **IT GRC > Administration > Conditions**.
2. Click **New**.
3. Populate these fields:
 - **Name:** Name of the condition collection.
 - **Description:** Description of the condition collection.
 - **Table:** Table on which the condition applies.
 - **Reference Field:** For supplemental conditions, the reference field for the table on which the primary condition is running.
 - **Condition:** Condition builder for defining the condition.

When performing a control test, processing dependencies are evaluated.

- If a control test definition is active, the system generates the control test instances dynamically, according to definition. To generate a control test manually:

1. Navigate to **IT GRC > Administration > Control Test Definitions**.

2. Open a control test definition record.

3. Click **Execute Now**.

ServiceNow generates a control test instance, marks it Pending, and assigns it to the group or individual responsible for the test according to the control test definition.

- If sample data was requested in the definition, any sample data that matches the conditions is found in the **Supporting Data** section. The **Test Complete Data Values** related list holds references to the records returned by the sample data query.
- If a control test has a condition type of **Basic**, the value in the **Sample size** field limits the number of failures that are stored as support data. If the result is passed or compliant, all the matching data is stored.
- If a control test has advanced conditions, the system evaluates them as follows:
 1. The condition collection in the **In scope** definition reference is processed.
 - a. The primary condition is processed on the fields specified in **Table** and **Fields** on the control test definition and returns an array of elements.
 - b. For each element in the array returned by the primary condition, supplemental conditions are processed, filtering the array of elements further.
 - c. The **In Scope** field is updated with the number of elements in the array.
 2. The condition collection in the **Configuration** reference is processed on the array of elements returned from the **In scope definition**. The choices for **Configuration to retrieve** are:
 - **None**: These conditions are skipped. **Supporting Data** includes all the elements that were in scope.
 - **Matching**: The control test checks the array of elements, returning any elements that match the **Configuration**.

- **Non-matching:** The control test checks the array of elements, returning any elements where at least one condition did not match the **Configuration**.

3. The final array of elements is recorded as **Supporting Data** records.

Remediation Tasks

If the control test reveals problems in the process, create a task from the **Remediation Task** related list. You can relate remediation tasks to any task in the system with the related items tool from the Many to Many Task Relations plugin.

Scripted audits

A scripted audit enables users with the certification_admin role to conduct an audit from a script rather than using restrictive template conditions.

A scripted audit uses a [certification filter](#) to select the records to audit, and then creates standard follow-on tasks for remediation of any discrepancies. Use this type of audit to query for any values or states that a script can define. A scripted audit is a specific audit type that is activated together with the [Desired State](#) plugin. ServiceNow provides a sample audit script with configuration instructions.

A scripted audit is an audit whose conditions are defined by a script.

1. Navigate to **All > Compliance > Scripted Audits > Audits**.

An audit type of Scripted filters the list.

2. Click **New**.

3. Complete the form (see table).

4. Create the audit script.

The Run this script field includes a sample script with instructions for performing the audit and generating the follow-on tasks. This field appears only when you access audits from the Scripted Audits module.

5. Click **Submit**.

Sample script:

```
/*
///////////////////////////////
/// This script works with Data Center Zones filter //
///////////////////////////////

var desiredFloorSpaceUsage = 30; // Value to audit against
var assignToUser = '46d44a23a9fe19810012d100cca80666';
// Beth Anglin
var assignToGroup = '8a5055c9c61122780043563ef53438e3';
// Hardware group
var taskMsg = 'See the audit results below for the discrepancies that must be addressed';

// API call to retrieve records based on the filter
var gr = new SNC.CertificationProcessing().getFilterRecords(current.filter);

// Loop over all records defined by the filter
while(gr.next()) {
    var sysId = gr.getValue('sys_id'); // Sys ID of audited record
    var floorSpaceInUse = gr.getValue('floor_space_in_use'); // Value to audit

    // Determine if certification condition passes or fails
    if (floorSpaceInUse < desiredFloorSpaceUsage) {
        var columnNameSpace = gr.floor_space_in_use.getLabel(); // String value of column audited against

        // Call create Follow on Task API and save the returned sys_id for use in logging audit result fail
        // Params:
        // auditId - Sys id of the audit record executed
        // ciId Sys - id of the configuration item. Empty string if not a cmdb ci
        // assignedTo - Sys id of user to assi
```

```
gn task to. Can be empty
                // assignmentGroup - Sys id of group t
o assign task to. Can be empty
                // shortDescr - Short description for
the Follow On Task. Can be empty
                // Return value: Sys id of the created
follow on task
                var followOnTask = new SNC.Certificatio
nProcessing().createFollowOnTask(current.sys_id, sysId
, assignToUser, '', taskMsg);

                // Call log failed result API
                // Params:
                // auditId - Sys id of audit record ex
ecuted
                // auditedRecordId - Sys id of the rec
ord audited
                // followOnTask - Sys id of the follow
on task associated with the audited record(see audit
edRecordId). Can be empty
                // columnDisplayName - Label of the co
lumn audited(ex. Disk space (GB)). Can be empty
                // operatorLabel - Label of the operat
or used to audit the column(ex. is not empty, greater
than). Can be empty
                // desiredValue - Desired value of the
column. Can be empty
                // discrepancyValue - Discrepancy valu
e. Can be empty
                // isCI - True, if audited record is a
CI. False, otherwise.
                // domainToUse - Sys domain of the "ce
rt_audit" record. Can be empty
                new SNC.CertificationProcessing().logAu
ditResultFail(current.sys_id, sysId, followOnTask, col
umnNameSpace, 'greater than', desiredFloorSpaceUsage,
floorSpaceInUse, true);
            } else { // If certification condition pass, wr
ite a Audit Result Pass via API
                // Params:
                // auditId - Sys id of audit record ex
ecuted
                // auditedRecordId - Sys id of the rec
ord audited
```

```
        // isCI - True, if audited record is a
        CI. False, otherwise. Can be empty.
        // domainToUse - Sys domain of the "ce
        rt_audit" record. Can be empty.
        new SNC.CertificationProcessing().logAu
        ditResultPass(current.sys_id, sysId, true);
    }
*/
```

New scripted audit table

Field	Description
Name	Name for this audit.
Filter	Filter to use when the audit type is Scripted. This field is required for scripted audits, but is hidden for all other audit types.
Template	[Required] Template to use when this audit runs. Audit type filters the list of available templates and only the active versions of a template are available for selection. This field is hidden when the audit type is Scripted.
Table	[Read-only] Displays the table for the template.
Create tasks	Creates follow-on tasks for correcting discrepancies when selected. In a scripted audit, you can create the logic for either task state by using true to create a task or false if no task is created. By default, this check box is cleared (false) in a new audit record.

Field	Description
Assignment type	<p>A choice list to select how the audit assigns the follow-on tasks. This field is visible only when the Create task check box is selected. Choices are:</p> <ul style="list-style-type: none">• User Field: elect a user reference field on the table being audited. As an example, select the user named in the Managed by field on the failed record to perform the tasks. This selection displays the Assigned to and Assign to empty fields. If the reference field on the record is empty, the value in the Assign to empty field is used.• Specific User: Select a specific user to perform the tasks. This selection displays the User field.• Group Field: Select a group reference field on the table being audited. As an example, select the Support group from the failed record to perform the tasks. This selection displays the Assign to group and Assign to empty fields. All members of the group from the reference field on the failed record are assigned to the tasks. If the reference field on the record is empty, the value in the Assign to empty field is used.

Field	Description
	<ul style="list-style-type: none"> Specific Group: Select a specific group to perform the tasks. This selection displays the Group field. All members of the selected group are assigned to the tasks.
User	<p>The specific user this audit assigns to follow-on tasks. This field is available under these conditions:</p> <ul style="list-style-type: none"> Assignment type is set to Specific User. Assign to empty is set to Create Assigned Task, and Assignment type is set to User Field. <p>Note: Ensure that the specified user has the certification role.</p>
Assign to group	The group field that defines which group this audit assigns to the follow-on task. This field is available only when the Assignment type is Group Field.
Group	The specific group this audit assign to follow-on tasks. This field is available only when the Assignment type is Specific Group and you have selected Group Field as the assignment type.
Assign to	The user field that defines which user this audit assigns to the follow-on task. This field

Field	Description
	<p>is available only when the Assignment type is User Field.</p>
Assign to empty	<p>The behavior to use if the field selected in Assign to or Assign to group is blank on the record being audited. For example, if a follow-on task must be assigned to a manager, but no manager is identified, the value in this field determines what happens. This field appears only when the Assignment type is User Field or Group Field. The possible selections are:</p> <ul style="list-style-type: none">• Do Not Create Task: No follow-on task is created when the Assign to or Assign to group field is empty.• Create Unassigned Task: Create a follow-on task, but do not assign it to any user or group. The task can be manually assigned later.• Create Assigned Task: Create a follow-on task and assign it to the user or group specified. If you selected an assignment type of User Field, the User field becomes available. If you selected the Group Field type, the Group field becomes available. <p>The audit automatically creates follow-on tasks for all records that have Assign to populated,</p>

Field	Description
	regardless of which selection you make for Assign to empty.
Short description	Brief description of the purpose of the audit.
Task description	General description of the work required for the follow-on tasks created by this audit. All follow-on tasks created by this audit inherit this description.
Active	Activates this audit schedule and generates follow-on tasks at the scheduled date and time. Clear this check box to hide scheduling fields on the form (except Last run date) and not generate follow-on tasks.
Run	<p>How often to run the schedule that generates the audit.</p> <ul style="list-style-type: none"> • Daily • Weekly • Monthly • Periodically • Once • On demand
Day	<ul style="list-style-type: none"> • If Run is Weekly, the day of the week when the audit runs. • If Run is Monthly, the day of the month when the audit

Field	Description
	runs. If the day is 29, 30 or 31, for shorter months the audit runs on the last day of the month.
Repeat Interval	If Run is Periodically, the frequency that the audit runs entered in time, days, or both. For example, set Days to 10 and Hours to 14:00:00 to run the audit every 10 days at 2:00pm.
Starting	If Run is Periodically or Once, the date and time when the audit runs.
Time	If Run is Daily, Weekly, Monthly, or Once, the time of day, on a 24-hour clock, when the audit runs.
Last run date	[Read-only] The last date and time the audit ran, either on its regular schedule or manually. Audit previews do not update this field.
Next scheduled run	[Read-only] The next date and time on which the audit runs. The system recalculates this field when you change the schedule.
Audit type	<p>[Read-only] The type assigned to this audit. The system selects the audit type based on the application from which the audit was created and can be:</p> <ul style="list-style-type: none"> • Desired State • Architecture Compliance

Field	Description
	<ul style="list-style-type: none">• Compliance• Scripted
Run this script	Audit script to run. This field is available only when the audit type is Scripted. The Audit form includes a sample script with instructions for performing the audit and generating the follow-on tasks. See Script Methods for a list of the methods provided and the accepted parameters.

ServiceNow provides four methods for creating the audit script.

Script methods

Name	Description	Parameters
getFilterRecords	public GlideRecord getFilterRecords(String filterId)	filterID: The sys_id of the filter to use.
logAuditResultPass	public void logAuditResultPass(\$String auditId, String auditedRecordId, boolean isCI, String domainToUse)	auditId: Sys_id of audit record executed auditedRecordId: Sys_id of the record audited. isCI: True, if the audited record is a CI, false if otherwise. domainToUse: Sys_domain of the cert_audit record.

Name	Description	Parameters
logAuditResultFail	<pre>public void logAuditResultFail(String auditId, String auditedRecordId, String followOnTask, String columnDisplayName, String operatorLabel, String desiredValue, String discrepancyValue, boolean isCI, String domainToUse)</pre>	<p>auditId: Sys_id of audit record executed.</p> <p>auditedRecordId: Sys_id of the record audited.</p> <p>followOnTask: Sys_id of the follow-on task associated with the audited record and can be an empty string.</p> <p>columnDisplayName: Label of the column audited. For example, Disk space (GB).</p> <p>operatorLabel: Label of the operator used to audit the column. For example, is not empty or greater than can be the label.</p> <p>desiredValue: Desired value of the column.</p> <p>discrepancyValue: Discrepancy value.</p> <p>isCI: True, if the audited record is a CI, false if otherwise.</p> <p>domainToUse: Sys_domain of the cert_audit record.</p>

Name	Description	Parameters
createFollowOnTask()	public String createFollowOnTask(St ring auditId, String cild, String assignedTo, String assignmentGroup, String shortDescr)	auditId: Sys_id of the audit record executed. cild: Sys_id of the configuration item. This string is empty when the table is not extended from the cmdb_ci table. assignedTo: Sys_id of the assigned user of the task. This string can be empty. assignmentGroup: Sys_id of the group the task is assigned to. This string can be empty. shortDescr: The text to use for the short description of the follow-on task.

Managing proposed changes

The proposed changes feature allows you to pre-configure changes to configuration items and their associated relationships. These pre-configured changes are prepared to be implemented, but do not actually happen until they are applied at a later time.

When you view a CI, the proposed changes can be displayed so that you can see what is planned.

This feature is useful when you want to make modifications while a change process is in the approval stage, and only implement the changes after the approvals are complete. If the change is never

approved, no changes to records have to be reversed. If the change is approved, a quick command applies all the proposed changes.

You can make the following proposed changes to a CI:

- Modify any field on the CI form.
- Add or delete a relationship to that CI.

To modify a relationship, you must delete the current relationship and add a new relationship. You cannot delete a proposed change.

View CI history

You can view the history of changes to a CI in a list, calendar, or timeline format.

View the proposed changes of a CI

You can view the proposed changes so that you can see what is planned for the CI.

Before you begin

Role required: personalize_form

About this task

To view any proposed changes, configure the CI form layout to display the **CMDB Scheduled Changes** field. Proposed changes are not displayed in a CI form by default.

Procedure

1. Navigate to **All > Change > Open** and open a change request.
2. In the **Affected CIs** related list, open the **Configuration Item**.
You may also navigate directly to the CI form.
3. Right-click the form header bar.
4. Select **Configure > Form Layout**.
5. Move the **CMDB Scheduled Changes** field to the **Selected** pane.

6. Click **Save**.

The CI form shows the details of any proposed changes in the **Scheduled changes** area.

Add a proposed change to a CI

Proposed changes to a CI can be made while viewing a change request or any task-related record.

Before you begin

Role required: itil

Procedure

1. In the Change Request form, go to the **Affected CIs** related list. If there are no CIs in the Affected CIs list, click **Edit** to add CIs that are affected by this change request.
2. Right-click the CI that you want to configure for a proposed change, and select **Proposed Change**.
3. Complete the form to make the proposed changes, and click **Save Proposed Change**.
Click **Update** to apply the changes immediately. Click **Delete** to delete the CI.
4. To propose an addition or a removal of a CI relationship:
 - a. Click the plus icon in the **Related Items** section.
 - b. In the Relationships section, add or delete a relationship. For information about using the relationship editor, see [Create or edit a CI relationship](#).
 - c. Click **Save Proposed Change**.
 - d. Confirm saving the proposed change.
Click **Update** or **Delete** to commit the changes immediately.

Note: Use only with CI relationships. Proposing additions or removal of relationships is not valid for user relationships and group relationships.

What to do next

After the proposed changes are saved, the **Apply Proposed Changes** button appears on the Change Request form. This button lets the user commit the proposed changes to the CI. Your business processes determine the appropriate time to commit the changes. The CI retains the existing data until the proposed changes are committed. However, users can see that changes have been proposed.

Apply a proposed change to a CI

When you apply the proposed changes, all the proposed changes for that change request are applied to the configuration item. You can apply proposed changes without verification, or if verification tests of the proposed changes have failed.

Before you begin

Role required: itil

About this task

After you apply the proposed changes, the **Scheduled changes** part of the form displays **No scheduled changes found**. You can configure proposed change verification rules which you can use to verify proposed changes before applying the changes.

Procedure

1. Navigate to the **Change Request** form.
2. Click the **Apply Proposed Changes** button.
You may have to right-click the form header and select the **Reload Form** option to see the changes.

Create or edit a proposed change verification rule

Ensure that proposed changes meet business requirements and do not introduce invalid data to the CMDB, create a rule that includes a script to verify the proposed changes.

Before you begin

Role required: asset or itil

About this task

When you configure proposed change verification rules for a CI, you have an option to verify that the proposed changes pass the verification test script in the rule. The verification test results are logged as passed or failed, and you can view the results. Running the verification test is not mandatory, and a failed verification test does not prevent you from applying proposed changes.

Procedure

1. Navigate to **All > Configuration > Change Verification > Proposed Change Verification Rules**.
2. Click **New** or select an existing rule to edit.
3. Fill in the fields, as appropriate.

Proposed Change Verification Rules form

Field	Description
Rule name	The name of this rule.
Table name	The table to which the rule applies.
Filter condition	Conditions to apply this rule to specific CIs.
Active	Check box to activate this rule.
Rule script	A verification Java script that needs to return true or false. For example: <pre>validateRule() { var os = current.getValue("os"); if (os == "Windows") { return true; } else { return false; } }</pre>

Field	Description
	<pre>e("os"); var cpu = current.getValue("cpu_count"); //Use current.getValue(fieldName) to get the proposed change value, eg. var os = current.getValue("os"); //Your verification code if (os != "SunOS" cpu < 2) return false; //Return true to pass the verification and false if the verification failed return true; }</pre>

4. Click **Submit** or **Update**.

Result

On the **Change Request** form, you can click **Verify Proposed Changes** to verify proposed changes for the affected CIs.

Verify proposed changes

Before applying proposed changes to affected CIs, use proposed change verification rules to verify that the changes meet business requirements and do not add invalid data to the CMDB.

Before you begin

Create or edit the rules used to verify proposed changes. For details, see [Create or edit a proposed change verification rule](#).

Role required: none

About this task

You can apply proposed changes even if they are unverified or fail a verification test.

Procedure

1. Open the **Change Request** form that affects the CI.
2. Click **Verify Proposed Changes**.
The proposed changes are verified against any proposed change verification rules in which the CI meets the **Filter condition** criteria.
3. Review the message that appears at the top of the form after the verification process is finished.
The message states whether the verification tests passed or failed.

What to do next

To view the details of any verification tests that were performed for the change request in the past two days, click the **Proposed Change Verification Log** related link.

Create or edit a planned change validation script

Create a custom script that checks if a change to a class was valid according to business requirements, and whether the change was planned or not. A planned change validation script is used whenever a CI change is viewed in the CI timeline or change history.

Before you begin

Role required: admin or itil

About this task

The system attempts to validate each CI change as follows:

- If a custom script exists for the CI or one of the CI parents, then the script is executed and the results are used to flag the change as valid or invalid. Parent CIs are examined in the hierarchical order.
-

If a custom script does not exist for the CI or any of its parents, then a predefined validation script is used. The change is determined as a planned change if the change occurred between the **Work start** and **Work end** dates of the change request associated with the changed CI.

However, this check is not always reliable because a user might have manually modified the CI within the work dates, which flags the change as valid even if it is invalid.

The script needs to return a boolean, true or false, which depends on meeting the test criteria in the script. You can define a separate script for each CI class, and you can define multiple planned change validation scripts for a single class. For example, to maintain different versions of the script. Only one script can be active for a CI class at any given time.

These are the parameters that uniquely characterize a change:

- The fields that were changed
- The data source that performed the change
- The time stamp of the change

To correctly determine the validity of a change, examine the parameters and apply business logic to evaluate if the validation tests are met. A planned change validation script can test any of these characteristics and determine when a change meets pre-established criteria. For example, the custom script can check if the mode of the CI is operational or maintenance, or who initiated the change.

Procedure

1. Navigate to **All > Configuration > Change Verification > Planned Change Validation Script**.
2. Click **New** or select a validation script to edit.
3. Complete the form.

Planned change validation script form

Control	Description
Active	Check box to activate this script for validating changes.
Applies to	Class that this script applies to.
Script	Script to run to validate a change. If the script does not return a boolean value, then it is configured to false.

The script has a template which displays the input variables of the script.

Template script input variables

Variable	Type	Description
current	GlideRecord	Current record that is being processed.
updatedOn	GlideDateTime	Time stamp of the change.
updatedBy	String	Entity responsible for the change.
fieldsChanged	String	Comma-separated list of the names of all fields that were changed.

This sample script checks who initiated the record update. It returns true if admin initiated the record update. Otherwise, the script returns false.

```
isValidChange();  
  
function isValidChange(/*GlideRecord current, GlideDat  
eTime updatedOn, String updatedBy, String changedField  
s*/) {
```

```
//Return true if the user that updated the record has an admin role
    return isUserAdmin(updatedBy);
}

function isUserAdmin(userName)
{
    var grUser = new GlideRecord("sys_user");
    grUser.addQuery('name', userName);
    grUser.query();
    if(grUser.next())
    {
        var roles = new GlideRecord ("sys_user_has_role");
        roles.addActiveQuery();
        roles.addQuery('user', grUser.sys_id);
        roles.query();
        while(roles.next())
        {
            if(roles.role.name == 'admin')
                return true;
        }
    }
    return false;
}
```

4. Click **Submit**.

Intelligent Search for CMDB

Use everyday natural language query (NLQ) in a search string to query for a set of CMDB objects. Intelligent Search for CMDB parses, resolves ambiguities, and converts your search string into a valid CMDB query. Complex search strings open fully constructed on a canvas of CMDB Query Builder where you can continue and refine, or run.

Intelligent Search for CMDB is supported only in English.

Integration with CMDB Workspace

Intelligent Search for CMDB is integrated into the CMDB Workspace store app. See [CMDB Workspace](#) for details about using Intelligent Search for CMDB in CMDB Workspace.

Integration with CMDB Query Builder

Intelligent Search for CMDB is integrated with the [CMDB Query Builder](#) in the Now Platform. This integration is controlled by the system property `glide.cmdb.query.nlq.activated`, which is set to **true** by default. Intelligent Search for CMDB lets you use natural language processing in the CMDB Query Builder to find CIs and their relationships using Intelligent Search for CMDB functions.

Using Intelligent Search for CMDB

Intelligent Search is tailored to the CMDB, searching only through the CMDB class hierarchy for tables, and for CIs and their relationships.

Use the Intelligent Search search field to construct a search string using everyday natural language. Your queries can span multiple CMDB classes and involve many CIs that are connected by different relationships. After resolving any ambiguities with table names or relationship types, Intelligent Search converts your search string into a query that the CMDB can run. The CMDB query is constructed dynamically as you type into the search box and spell checker is applied if needed. A dynamic list of relevant suggestions appears as you type, with items such as table names, matching single words or part phrases in the typed-in text.

Use Intelligent Search:

- **Search tips:** Shows details and tips about the usage, and examples for single and multi-table search, advanced filtering, and relationships in Intelligent Search. The Relationships tab contains a link to the [CMDB Implicit Relationships](#) table.

•

Search: Depending on whether the search string is already fully converted into a valid CMDB query and whether the search is for a single or multiple tables.

•

If the search string has no ambiguities with the table name or relationships, then the query runs and the results appear in a list view format.

If the constructed CMDB query contains more than a single table, then the **View in Query Builder** button appears. Click the button to open the **CMDB Query Builder** with your query fully constructed on the Query Builder canvas. You can use the Query Builder to continue editing the query.

- If there are any ambiguities with table names or relationship types in the search string, then the search string can't be converted into a valid CMDB query. In this case, the Refine your query dialog box appears to continue and parse your search string into a valid CMDB query. The dialog box contains suggested synonyms and labels for phrases in your search string. Use the drop-down lists to select the synonyms that match your intended search and then click **Go** to run the query.
- If Intelligent Search is unable to convert your search string into a valid CMDB query, then clicking **Search** does not generate any query results. Instead, a feedback form appears. Fill out the form and click **Submit Feedback** to send your feedback to ServiceNow analysis.
- **Results Feedback:** Submit feedback to ServiceNow analysts, to express your assessment of the results. Choose descriptions that capture any gap between the results and your expected results, and add any helpful details.

Sample searches

When you click the search box, the drop-down list of pre-defined sample searches appears. The list consists of more common searches, or searches that are more difficult to construct such as searches that involve application services. Run any of those searches to get started.

- Sample searches are stored in the NLQ Sample Search [sn_cmdb_ws_nlq_sample_search] table
- Referenced tables are stored in the NLQ Sample Search Table [sn_cmdb_ws_nlq_sample_search_table] table

CMDB Admins (sn_cmdb_admin user role) can modify a sample search by directly editing its record in the NLQ Sample Search table. Click **All** and then in the Filter navigator, enter `sn_cmdb_ws_nlq_sample_search.list`. In the NLQ Sample Searches list view edit the record for a search that you want to modify.

Any modification to sample searches is reflected in both, the CMDB Workspace and the CMDB Query Builder.

Synonyms

The NLQ Synonym [nlq_synonym] table is pre-populated with synonyms for natural language strings for CMDB table and column names, and relationships. This table is used to match natural language search words to the CMDB query language. For example, the phrase 'linux server' has synonyms such as 'Linux Server', 'Server', and 'Virtual Machine Instance'.

For details about viewing and adding synonyms customized to your business needs, see [NLQ synonyms](#).

CMDB Implicit Relationships

You can help Intelligent Search find more results by defining some of the relationships between classes as implicit relationships. Implicit relationships can be useful in queries that involve service offering and application services.

NLQ admins can create new implicit relationships by navigating to **All > NLQ > CMDB Implicit Relationships**.

An implicit relationship defines the relationship between two tables and includes any filters you want to apply. When creating an implicit relationship, you set the following items:

- From table (from_table): The class that acts as the parent
- Filters: Conditions that are applied to the columns of the from_table
- To table: The class that acts as the child
- Relationship: How the from_table interacts with the to_table. For example, Contains: Contained by means the from_table contains the to_table
- Skipped table: The class that is implied and not captured by the CMDB Query Builder

For example, in CMDB Query Builder, you want to see your service offerings that have had a P1 incident in the last 10 days. However, if you were to type `show me all business service offerings with`

p1 incidents in the last 10 days, NLQ wouldn't understand the relationship.

Implicit relationships are stored in the NLQ CMDB Implicit Relationship [nlq_cmdb_implicit_relationship] table and are used in the CMDB Workspace and if integrated, also in CMDB Query Builder.

For more information about NLQ in the Now Platform, see [Natural Language Query \(NLQ\)](#).

Unified Map

The Unified Map feature graphically shows a hierarchical map of CIs and the relationships between them, while centered on a CI that you choose (referred to as the home node). The Unified Map feature combines some of the capabilities of Dependency Views and of Service Mapping, into a single map experience.

Unified Map is available starting with CMDB Workspace v4.0.

Unified Map visually shows how CIs are connected to each other, which lets it be useful with products such as [Change Management](#), [Incident Management](#), and [Event Management](#). Nodes on the map represent CIs in the CMDB hierarchy and different types of lines between the nodes represent the connections between CIs.

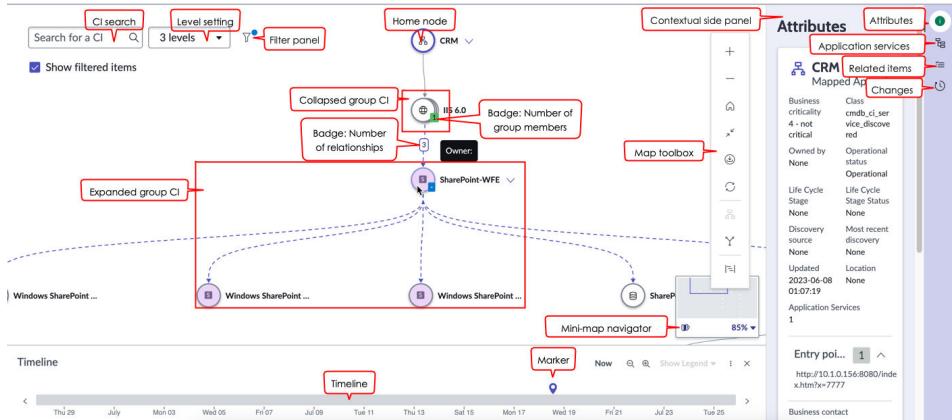
For example, Unified Map helps you understand the impact of a change by visually showing how CIs are connected to other CIs by relationships and references. Unified Map also shows the composition of application services which lets it be useful with products such as Event Management and Incident Management. For example, you can see all CIs that are members of the 'Revenue App' application service. You can review historical changes, and then easily filter the CIs so that only Application CIs appear.

Unified Map provides the following panels:

- Map pane that shows the map for a CI, and which lets you access the following utilities:
 - CI search field letting you [search for the home node CI](#) for the map
 - Filter panel letting you [filter CIs](#) and relationships that appear on the map, and [create filter presets](#) to reapply a set of filters on any map

- Toolbox letting you modify various visual aspects, such as the layout mode, of the map
- Mini-map navigator letting you easily move the entire map to an area of interest and set the zoom level for the map
- Timeline showing related item events for the CIs on the map, up to six months in the past and six months into the future, as of the time of viewing the timeline.
- Contextual side panel on the right where you can choose the **Attributes**, **Application services**, **Related items**, or **Changes** module. The panel shows different types of details per module and according to the selections on the map

Panels and elements in Unified Map



The map content and the details that appear in the various modules in the contextual side panel, is synchronized. The selected module in the contextual side panel determines the type of details that appear and your selection on the map determines the scope. If you select a CI on the map, then the contextual side panel shows details for that CI. For some modules, if nothing is selected on the map, the contextual side panel shows details for all the CIs on the map.

For example, if the Application services module is selected and a CI is selected on the map, then the contextual side panel shows all application services related to the selected CI. If you then select an empty space on the map so that no CI is selected — The contextual side panel shows all application services that are associated with any CI in the map.

Access

You can access the Unified Map feature from [CMDB Workspace](#), in either of the following ways:

- Navigate to **Workspaces > CMDB Workspace**. Then, in the Quick Links section on the CMDB Workspace Home view, select **Unified Map**.
- Select **Open Map** on a CI form, to open a map for the respective CI as the home node of the map.

Role requirements:

- To access maps: sn_cmdb_user, sn_cmdb_editor or sn_cmdb_admin roles
- To access maps with operational application services: app_service_user, and sm_user or sm_admin
- To access maps with operational and non-operational application services: app_service_admin, and sm_user or sm_admin
- To access and view related items: itil

Map content

The map pane shows a graphical layout of CIs and their relationships including group CIs such as application services, and a timeline. You can set and change the content on the map as follows:

Home node

Use the Search box to select the CI, such as an application service CI, that you want the map to center on. That CI is the home node of the map. The home node is easily visible by its thicker border line and by a pulsating effect. The home node is the focal point of a map and all other CIs and relationship lines are drawn in the context of the home node.

Level

Select the levels drop-down list to set the map level. The map level limits the number of hierarchy levels for which to show CIs, therefore, limiting the overall size of a map. Starting with the home node, the map shows all CIs up and down the hierarchy, up to the specified level. By default,

only the first three levels of CIs, directly descending from the home node, appear on the map.

The level setting has no effect when an application service CI is set as the home node (CI is a mapped application service). All levels of relationships appear for an application service CI.

Filter

Select Open filter panel () to filter the data appearing on the current map, and create [filter presets](#). Each filtering category includes only items that are relevant to the current map. The list in the CI types category, for example, includes all the CI types on the map, and you can filter in or out any of those items. A blue dot appears on the filter icon when filters are being applied.

Filtered items don't appear on the map unless you select **Show filtered items** to expose them as grayed-out CIs and relationships.

For more information about configuring map filters, see [Configure map filters](#).

Service Mapping

When [Service Mapping](#) is installed then application services populated by Service Mapping population methods, are included in maps. Otherwise, only application services populated by CMDB-related population methods, appear.

Without Service Mapping, any other data that Service Mapping provides for application services, such as relationships, grouping, and mapped application services data (and subclasses of that class), isn't available. Also, the Application services module in the contextual side panel, doesn't show details for CIs in the map.

Note: A map retrieves and shows only up to 250 CI nodes (CIs within a collapsed group CI are counted). Any additional elements are truncated and don't appear on that map.

Map appearance and interaction

The map initially appears using default settings. Colors used in Unified Map to highlight some field values, follow the Next Experience color themes, and dashed lines represent references between end point CIs.

You can interact with the map and change the appearance of its current content, as follows:

- Use the toolbox on the map to change map appearance:
 - Zoom in or out: Add or reduce the level of details on the map.
 - Align to home node: Move the home node to the center of the map.
 - Fit to screen: Center the map on the canvas and set the zoom level to its maximum level that enables the entire map to fit on the canvas.
 - Export map: Export the map to a PDF document on your local drive.
 - Collapse all: Collapse all group CIs that are expanded, center the map on the canvas, and set the zoom level to its maximum level that enables the entire map to fit on the canvas.
 - Switch map layout to either of the following options:
 - Unified vertical layout: Shows elements in a vertical tree pattern according to their upstream and downstream relationships. This option is the default layout for showing mapped application services.
 - Unified force layout: Shows elements in a clustered arrangement around a parent CI, regardless of upstream or downstream relationships.
 - Toggle visibility of the timeline: Show or hide the timeline underneath the map. For more information about the timeline, see [Show CI related items events on a timeline](#).
- Point to a CI to animate the relationships between the CI and its connected CIs.
- Use the mini-map navigator on the bottom right to easily move the entire map on the canvas and set the zoom level to a specific percentage number. You can hide or show the mini-map navigator.

- Select the number badge on a group CI to expand the group and show its member CIs. The group parent and any CI members are temporarily highlighted by a light purple color (point to a group if needed). Select the badge again to collapse the group back into its group CI mode.
- Drag CI nodes on the map canvas to reposition CIs as needed.

Contextual side panel

Select one of the following modules in the contextual side panel on the right to show different types of details for CIs on the map:



- Attributes (): Shows the Attributes pane with details that are based on selections in the map. For more information about using this module, see [Show attributes for a CI or a relationship in a map](#).
-



Application services (): Cards in the Application services pane show key details, such as Owner and Discovery source, for application services. If a CI is selected on the map, then application services related to the selected CI, appear. If nothing is selected on the map, then any application service that is related to any CI on the map, appears.

For more information about using this module, see [Show application services for a CI in a map](#).

-



Related items (): Shows related items grouped by related items categories, such as active incidents and alerts, in the Related items pane. If a CI is selected on the map, then only related items that are associated with the selected CI, appear. If nothing is selected, then all related items that are associated with any CI on the map, appear.

- For information about using this module, see [Show related items for a CI in a map](#).

- For information about how administrators can configure the related items details appearing in badges, on timelines, and in cards in the Related items panel, see [Configure options for the Related items module](#).

-



Changes (): Shows CI changes in the Changes pane. If a CI is selected on the map, then only changes for the selected CI, appear. If no CI is selected, then the Changes pane doesn't show any change details.

For more information about using this module, see [Show changes for a CI in a map](#).

Service maps

Mapped services, appear on the map as group CIs that you can expand and collapse to show or hide its members. Mapped services are descendants of the Application Services [cmdb_ci_service_auto] class, such as application services and dynamic CI groups. A badge on a group CI shows the number of members in the group.

Select the badge to expand the group to show all members, and then collapse the group back to hide members and to show only the group CI. In its expanded mode, point to any CI in the group to temporarily highlight all CI members by a light purple color.

Administer Unified Map

As an administrator, you can modify several settings in Unified Map to reflect how maps are used in the organization.

The Unified Map configurations in the following tasks affect all users. Users can customize some aspects of their own experience with Unified Map, but can't configure settings in these tasks.

Add related items categories and configure what appears on individual cards in the Related items module, and how related items appear on timelines and in badges.

Before you begin

Role required: sn_cmdb_admin

About this task

Related items are grouped by categories in the Related items pane in the contextual side panel, and also appear in CI badges and in timelines. The Node Map Related Item [sn_cmdb_ws_node_map_related_item] table contains the settings that determine which related items and associated details, appear. By default for example, common categories of related items, such as active incidents, are pre-configured to appear. You can add or modify records in the Node Map Related Item [sn_cmdb_ws_node_map_related_item] table to globally manage related items in maps.

Procedure

1. Navigate to **All** and then, in the Filter box in the main navigation bar, enter `sn_cmdb_ws_node_map_related_item.list` to open the Node Map Related Item table.
2. Select an existing record or select **New** and then fill out the form.

Field	Description
Name	The category label that appears in the contextual side panel when the Related items module is selected for a CI.
Table	Tables from which records for the category are retrieved.
Order	Order that the category appears within all related items categories. The list of related items categories is sorted in an ascending order. The category with the smallest order number is at the top of the list.

Field	Description
Active	Enables the appearance of the related item category.
Reference field	Reference attribute in the specified Table that references the CIs for the category. Typically set to Configuration Item [cmdb_ci] .

Configure related items in the Related items contextual side panel.

Related fields and conditions tab

Number field	Numeric attribute from the specified Table that uniquely identifies each record in the category. This attribute is used in the record links that appears in each individual card when drilling down the related item category.
Title field	Attribute from the specified Table that appears as the title of each individual card when drilling down the category in the contextual side panel.
Fields	Set of attributes from the specified Table that appear in individual cards when drilling down the related item category. For example, when drilling down alerts, the set of attributes that appear on each individual alert card.
Sort fields	Order of appearance of the specified set of Fields .

Footer field	Attribute from the specified Table that appears at the bottom of individual cards when drilling down the related item category, regardless of the sort order specified in Sort fields .
Conditions	Conditions to apply to the specified Table that retrieve the set of records for the category.

Configure related items in badges and timelines.

Badge and timeline configuration tab

Badge and timeline icon	Icon that appears in CI badges and on timelines, for the specified related item.
Badge and timeline highlight field	Highlight configuration for applying colors to the icons in timelines and on the map.
Date field	The date field to position the event on the timeline.
End date field	Optional end date field to use for a range on the timeline.

3. Select **Submit** or **Update**.

Each class has a unique set of extended properties that appear in the Unified Map Attributes panel for a CI. Many common classes are pre-configured with such set of extended properties. You can modify these default settings and globally configure extended properties for additional classes.

Before you begin

Role required: sn_cmdb_admin

Procedure

1. Navigate to **All** and then, in the Filter box in the main navigation bar, enter `sn_cmdb_ws_node_map_table_attributes.list` to access the Table Attributes table.
2. Select an existing record or select **New** and then fill out the form:

Field	Description
CMDB Class Name	The class that this configuration applies to.
Display attributes	<p>List of attributes that appear as extended attributes in the contextual side panel when a user selects the Properties module for a CI. The list contains attributes of the specified CMDB Class Name from which you can select.</p> <p>If the field is locked, then to configure, select the Unlock Display attributes icon first.</p>

3. Select **Submit** or **Update**.

Configure map references to connect on the map, CIs from two classes that aren't connected by a relationship.

Before you begin

Role required: sn_cmdb_admin

About this task

References connect CIs from two classes that don't have a relationship connection between them. On the map, any two CIs from the referenced class and from the referencing class, appear connected by a dotted line as if there's a relationship connection between them. For such reference connections, the relationship type is **Reference**. There are several pre-configured map references which you can modify, and you can also add additional map references.

For example, a Windows Server CI record has a map reference to records in the File System table that you want to include in a map. In that case, Server ABC (referencing CI) shows connections to C:\ and D:\ file systems (referenced CIs).

Procedure

1. Navigate to **All** and then, in the Filter box in the main navigation bar, enter `sn_cmdb_ws_node_map_reference.list` to open the Node Map References table.
2. Select an existing record or select **New** and then fill out the form.

Field	Description
CI class	The referencing class that this configuration applies to.
Referenced CI class	Class that the specified Reference field references.
Reference field	Attribute in the specified CI class that contains the reference.
Show Reverse	Creates a map reference for the reverse reference between the specified CI class and Referenced CI class .
Active	Enables the configuration so that the map reference shows referenced CIs.

3. Select **Submit** or **Update**.

Configure profiles that set default map filters and default map orientation for a class. For example, to show Service Mapping data for the Mapped Application Service class.

Before you begin

Role required: sn_cmdb_admin

About this task

Class profiles are applied when no [filter preset](#) is used with the current map. This typically happens when you initially load a map without a filter preset or when you set the filter preset to **Default view**. Class profiles let you configure only the Layer category in the filter panel. Several common classes such as the Mapped Application Service [cmdb_ci_service_discovered] class, are pre-configured with class profiles.

Procedure

1. Navigate to **All** and then, in the Filter box in the main navigation bar, enter `sn_cmdb_ws_node_map_profiles.list` to open the Node Map Profiles table.
2. Select an existing record or select **New** and then fill out the form.

Field	Description
Orientation	Orientation of the map.
CMDB CI class	Class of the home node of the map for which to apply this profile.
Layers	Filters in the Layers category, to filter elements in or out of the map.
Active	Designates whether this profile is active.

3. Select **Submit** or **Update**.

Show a CI in a map

Select a CI as the home node that a map centers on. The home node in Unified Map is the focal point of a map. All other CIs and relationships on the map are drawn in the context of the home node.

Before you begin

Role required: See general role requirements for Unified Map

About this task

When you open a map for an application service CI, a badge on group CIs in the map shows the number of CI members in that group. In that case, group functionality is available letting you expand or collapse the group CIs.

Procedure

1. Navigate to **Workspaces > CMDB Workspace**.
2. In the Quick links section in the Home view, select **Unified Map**.
3. In the search field, search for a CI that you want to see a map for.
The CI that you find and select, is automatically set as the home node for the map.

What to do next

You can change the map content or appearance as described in [Unified Map](#), and you can do any of the following interactions:

- Show CI key details on the map: Point to a CI to show the CI's full name and class (or zoom into the map until those details appear), its related items details, and the direction of the CI relationships. If there are multiple related items associated with a CI, then the CI badge contains the string 'Multiple' and a badge showing the number of related items appears on the timeline. You can point to that badge on the timeline to show all related items.

- Show complete CI details: Select a CI (CI border thickens) to show its respective details in the Attributes, Application services, Related items, or Changes module panes.
- Show relationship details: Select a line, which represents relationships between the CIs, to show the relationship attributes in the Contextual side panel on the right (select the Attributes module if needed). If there are multiple relationships between CIs, then a badge shows the total number of relationships, and details for all the relationships appear in the Attributes panel.
- Move a CI: Drag a CI to a different place on the map. Map elements that are connected to that CI, are automatically moved and redrawn according to the new placement of the CI.
- Change the level setting: Select the level drop-down and set the number of relationship levels from the home node, for which to show CIs on the map.
- Select a different home node: Right-click a CI. Then, in the pop-up menu, select **Set as Home node** to redraw the map with the selected CI as its home node.
- Expand and collapse a group CI: Select the number badge (indicating the number of CI members) on a group CI to expand the group and show its CI members. Select the badge again to collapse the group and to show the group CI and its number badge. In its expanded mode, the group parent and any CI members are temporarily highlighted by a light purple color (point to a group if needed).
- Show filtered items: Select **Show filtered items** to expose the filtered items on the map, in a grayed-out shade. For more details about configuring map filters, see [Configure map filters](#).

Show CI related items events on a timeline

Show related items events, such as incidents and changes, associated with the CIs on the map, across a time range.

Before you begin

Role required: See general role requirements at [Unified Map](#)

About this task

Select a CI on the map to show its related items on the timeline, or select an empty space on the map to show related items for all of the CIs on the map. Multiple related items from the same date are grouped into a badge which shows the number of related items in the group.

The time range of timelines stretches from six months in the past through six months into the future, relative to the current time. An event that occurred outside of this time range, doesn't appear on the timeline. However, such events might still appear in the corresponding Related items pane and a CI badge might show data for those related items.

A timeline uses a marker to filter out CIs that were created after the marker date, and doesn't affect related items on the timeline. Set the marker to visualize topology changes in the CMDB until a specific point in time. Maps are synchronized with their associated timelines, showing the CIs and relationships as they existed at the marker's date and time. For example, CIs that were created after the marker's date and time, don't appear on the map.

As an administrator, you can configure some properties of the timeline, such as which related items details appear on timelines. For more information about configuring timelines, see .

Show related items events in a timeline, as follows:

Procedure

- Point to the marker to show its date setting.
- Point to a related items badge to show details about the related items for that point of time.
- Drag the marker to another date to be set as a marker.
- Zoom in or out to extend or to shrink the range of dates that show on the timeline.
- Select the More options button and then select **Replace marker**, **Clear marker**, or **Add marker** to manage the marker.
- Select **Now** to set the marker to current time.

What to do next

Use the map toolbox to show or hide the timeline.

Configure map filters

Specify the type of CIs that you want to see on the current map by filtering out data using different filtering categories.

Before you begin

Role required: See general role requirements at [Unified Map](#)

About this task

Some maps might extend to be too large to be useful. After selecting the home node, you can reduce the size of a map by filtering out CIs of specific type or with specific attribute values. For example, you can filter out CIs associated with a specific discovery source or which are owned by a specific user, and which you aren't interested in.

The Map filter panel lets you filter by layer types, CI classes, and relationship types. Filter options reflect the elements that are currently on the map and represent attribute values from all elements in the map. Select or clear items to filter CIs and relationships in the map.

Note:

- Filter conditions that might filter out the home node CI might be available in the map filter panel because of other CIs on the map. However, you can't filter out the home node CI even if it matches those filter conditions.
- Maps retrieve and show only up to 250 CMDB elements and any remaining elements are truncated and don't appear on the map.

Procedure

1. Navigate to **Workspaces > CMDB Workspace**.
2. In the Quick links section in the Home view, select **Unified Map**.

3. In the search bar, search for the CI you want as the home node for the map.
4. Select Open filter panel ().
5. In the Map filter panel, select or clear filters that you want to apply to the current map.
6. Close the Map filter panel.

What to do next

- Select or clear **Show filtered items** to expose all filtered items as grayed-out elements, or to hide them.
- Create a [filter preset](#) to reuse a custom set of filters.

Manage filter presets

Create and save a custom set of Unified Map map filters, which you can then apply to any map to show only what you are interested in.

Before you begin

Role required: See general role requirements for Unified Map

About this task

After you [configure a custom set of map filters](#), you can capture those settings as a filter preset. You can then select that filter preset to apply automatically all of its filter settings to the current map.

Procedure

1. Navigate to **Workspaces > CMDB Workspace**.
2. In the Quick links section in the Home view, select **Unified Map**.
3. In the search bar, search for the CI you want as the home node for the map.
4. Select Open filter panel ().

5. In the Map filter panel, select the filters that you want to include in the filter preset.
The filter settings are also applied to the current map.
6. Select the star-shaped icon, enter a **Preset name**, and then select **Save**.
7. Close the Map filter panel.

What to do next

- Apply a filter preset: Select the View preset list icon In the Map filter dialog box, then select a preset filter and close the Map filter panel.
- Update a filter preset: Select a filter preset in the Map filter dialog box and update the filter settings. Then select the star-shaped icon, and in the Save Preset dialog box do either of the following:
 - Select **Update preset**
 - Select **Save as new preset** and enter a filter preset name
Select **Save**.
- Remove a filter preset: Select the filter preset in the Map filter dialog box. Without changing the preset, select the star-shaped icon, and in the Remove preset dialog box select **Remove**.

Show attributes for a CI or a relationship in a map

Open the Attributes module in Unified Map to show attributes for a CI and for relationships that are on a map.

Before you begin

Role required: See general role requirements for Unified Map

Procedure

1. Navigate to **Workspaces > CMDB Workspace**.
2. In the Quick links section in the Home view, select **Unified Map**.
3. In the Search bar, search for a CI to be set as the home node for the map.

4. Select a CI or a relationship on the map for which you want to show attributes.



5. Select Attributes () in the contextual side panel to access the Attributes module.

The Attributes pane shows the following details:

- For a non-group CI: The top section contains common key attributes, such as Class and Discovery source that appear for CIs of any class (other than Application Service CIs). The bottom section contains attributes that uniquely extend the CI's class in the CMDB hierarchy.
- For a group CI: The top card shows attributes of the group CI and underneath, the Configuration items section contains cards showing attributes for each of the group member CIs.
- For a single relationship: A card with the attributes for the selected relationship.
- For a multi-relationship badge: A card for each of the relationships in the relationship set, showing the attributes for the relationship. The number of cards is equal to the number on the relationship badge, which is a count of the different types of relationships between the two CIs, in the same direction. Relationship connections are based on records in the CI Relationship [cmdb_rel_ci] table.

What to do next

In a CI card in the Applications pane, select the Actions menu and then select any of the following options:

- **View CI details** to open the CI form for the CI.
- **Open in new map** to open an additional map with the current CI set as the home node.
- **Set CI as home** to set the selected CI as the new home node for the map.

Select a mapped application service CI on the map to show its entry points in the Attributes pane.

Show application services for a CI in a map

Open the Application services module in Unified Map to show details of application services associated with a CI that's on the map.

Before you begin

Service Mapping must be installed to show application services for a selected CI.

Role required: See general role requirements for Unified Map

Procedure

1. Navigate to **Workspaces > CMDB Workspace**.
2. In the Quick links section in the Home view, select **Unified Map**.
3. In the Search bar, search for a CI that to be set as the home node for the map.
4. Select a CI on the map for which you want to show application services.
5. Select Application Services () in the contextual side panel to access the Application Services module.

What to do next

- Select another CI on the map to show its related application services. Or, select an empty space on the map so that no CI is selected, to show all application services for any of the CIs on the map.
- In the Application services pane:
 - Use the Search box to search for a specific application service.
 - Select () to open the filter panel, and then select a category and a sort order by which to sort the application service cards. Close the filter panel to apply your settings.

In an individual application service card, select Application Service actions (⋮) and then select:

- **View CI details** to open the CI form for the application service CI.
- **Open in new map** to open a new map in which the application service is set as the home node.
- **Set CI as home** to set the application service as the home node on the map and redraws the map accordingly.

Some fields, such as Business criticality, are color-coded to denote specific values. For example, a status of Most Critical is highlighted by default with a red background.

Show changes for a CI in a map

Open the Changes module in Unified Map to show historical changes for a CI that's on the map.

Before you begin

Role required: See general role requirements for Unified Map

Procedure

1. Navigate to **Workspaces > CMDB Workspace**.
2. In the Quick links section in the Home view, select **Unified Map**.
3. In the Search bar, search for a CI to be set as the home node for the map.
4. Select a CI on the map for which you want to show changes.
5. Select Changes (⌚) in the contextual side panel to access the Changes module.

What to do next

- Select another CI on the map to show changes for.
- In the Changes pane:

- Select Open filters () and then select and configure any of the following filter types:
 - **Post types**
 - **Field changes**
 - **Flagged**
 - **Filter sets**
- Select the Search icon () and enter text in the Search Activity Stream field to find specific changes by entering terms.
- Select the  or the  sorting icons to toggle between sorting the change cards in an ascending or descending order.
- Select **Show less** or **Show more** to show the minimum or the maximum level of details in a change card.
-

In an individual card: Select the Flag as important icon to flag the change as important.

Show related items for a CI in a map

Open the Related Items module in Unified Map to show all related items such as active incidents or active problems, for a CI that's on the map.

Before you begin

Role required: See general role requirements for Unified Map

Procedure

1. Navigate to **Workspaces > CMDB Workspace**.
2. In the Quick links section in the Home view, select **Unified Map**.
3. In the Search bar, search for a CI to be set as the home node for the map.

4. Select a CI on the map for which you want to show related items.

5. Select Related items () in the contextual side panel to access the Related items module.

Related items are grouped by categories. The categories that appear are based on the [Related Items options](#) and badges show the number of related items of the respective category, for a selected CI (or for all CIs if no CI is selected).

What to do next

- Select another CI on the map to show related items for, or select an empty space on the map so that no CI is selected, to show all related items for any of the CIs on the map.
- Select Related items user preferences () in the Related items pane, to [set the related item categories](#) that you want to see in the Related items pane.
- Select a related item category (the category must contain least one item) to drill down to the individual related item cards and see key details for each related item.
- In a related item card:
 - Select the related item link to drill down to the actual related item record.
 - Open the filter panel to set a sorting category and order for the cards.

Some details, such as Priority and Risk, are color-coded to highlight certain values. Color-codes use the default ServiceNow platform color code settings.

Configure options for the Related items module

The Related items module in Unified Map lets you set your preference for which categories of related items, such as active incidents and active problems, appear for CIs.

Before you begin

Role required: See general role requirements for Unified Map

About this task

Related items are grouped by categories in the contextual side panel when selecting the Related items module. Administrators configure which related item categories are globally available, and you can further customize these settings to reflect your own preferences.

Procedure

1. Navigate to **Workspaces > CMDB Workspace**.
2. In the Quick links section in the Home view, select **Unified Map**.
3. Select the Related items icon  to open the Related items module.
4. In the Related items panel, select the Related items user preferences icon .
5. In the Related items Settings dialog box, select and deselect items in the Available items list to reflect which related item categories you want to show for CIs.
Selected items are moved into the Selected item lists and when you select a CI on the map for which to show related items, those categories appear.
6. Select **Apply**.

Dependency Views

ServiceNow® Dependency Views graphically displays an infrastructure view for a configuration item (CI) and the application or business services that it is part of and that it supports. Dependency Views indicates the status of its configuration items, and allows access to CIs related alerts, incidents, problems, changes, and services.

If Service Mapping is activated, Dependency Views maps are enhanced to display dependencies that reflect connections in service maps.

		Use
	Administer	<ul style="list-style-type: none">• Create or modify map indicators• Create or modify map icons• Create a predefined filter• Create or modify Map Related Items• Create or modify Dependency Views menu actions• Create or edit a dependency type
Explore		<ul style="list-style-type: none">• View a Dependency Views map• Change the layout of Dependency Views map• Filter the view of a Dependency Views map• Perform actions on nodes in a Dependency Views map• Supported browsers for Dependency Views
Develop	Integration	Troubleshoot and get help
<ul style="list-style-type: none">• Developer training• Developer documentation• Properties for Dependency Views	<p>View metrics for CIs in a Dependency Views map</p>	<ul style="list-style-type: none">• Ask or answer questions in the Now Community• Search the Known Error Portal for known error articles

- Components installed with Dependency Views
- Contact Customer Service and Support

Supported browsers for Dependency Views

The latest version or service pack of internet browsers are required to view and manipulate Dependency Views maps.

The Dependency Views module supports the latest version or service pack of the following browsers:

- Firefox with the latest ESR
- Chrome latest version
- Safari version 8 or later (latest is recommended)
- Microsoft Internet Explorer (IE) version 11 and Microsoft Edge.

The Dependency Views module is not supported on tablets and on mobile devices.

Domain separation and Dependency Views

Domain separation is unsupported in Dependency Views. Domain separation enables you to separate data, processes, and administrative tasks into logical groupings called domains. You can control several aspects of this separation, including which users can see and access data.

Support level: Basic

- Business logic: Ensure that data goes into the proper domain for the application's service provider use cases.
- The application supports domain separation at run time. The domain separation includes separation from the user interface, cache keys, reporting, rollups, and aggregations.
- The owner of the instance must set up the application to function across multiple tenants.

Sample use case: When a service provider (SP) uses chat to respond to a tenant-customer's message, the customer must be able to see the SP's response.

For more information on support levels, see [Application support for domain separation](#).

How domain separation works in Dependency Views

Dependency views are generated using both Configuration Item [cmdb_ci] and CI Relationship [cmdb_rel_ci] tables. The [cmdb_ci] table is domain separated, but the [cmdb_rel_ci] table is not. You can create relationships only by selecting two CIs. They should be in the same domain for you to be able to see them.

To be successful with domain separation in Dependency Views, make sure that relevant CIs are visible for the current domain. If the instance is domain separated, ServiceNow domain separation rules apply (see Related information link below).

Tenant domains will be able to see only their domain and global CIs.

Dependency Views map

ServiceNow® Dependency Views maps graphically display CIs that support application or business services and the relationships between the CIs.

The [CMDB Workspace](#) store app provides the [Unified Map](#) feature as an alternative to using Dependency Views. Unified Map combines the capabilities of Dependency Views and [Service Mapping](#) into a single map experience.

A ServiceNow service (application service or business service) is work or goods that are supported by an IT infrastructure. For example, delivering email service to an employee can require services such as email servers, web servers, and the work to configure the user's account.

A Dependency Views map has one starting point, called the root CI or root node of the map. The root CI is surrounded by a darker frame that repaints itself with a pulsing effect drawing the attention to the root CI. The maps can show both upstream and downstream dependencies for the root CI. By default the Dependency Views map displays 3 levels, both

upstream and downstream relationships. Administrators can configure the number of levels displayed. The map collapses and expands clusters to make them easier to view. By default, clusters are collapsed.

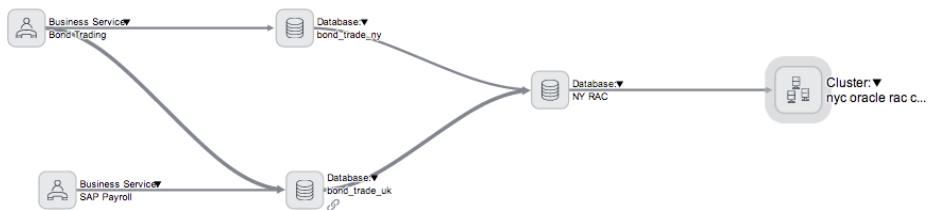
In a Dependency Views map, map indicators indicate if a CI has any active, pending issues. You can investigate the tasks that are connected to a CI to get more details. When you return to the map from another form, the system restores the last map viewed, using the default filter and layout settings. When you click the icon () on a CI record or on a task record that identifies a CI, the map opens.

Many of the relationships in map are created through the discovery process. You can also create, define, and delete CI relationships in the map. You can display the map from different perspectives and open specific records that relate to configuration items. The system refreshes the map automatically to reflect changes to the CMDB.

Note: CIs not extended from the Configuration Item [cmdb_ci] table, are not displayed in Dependency Views maps and in CI relation formatters.

The Dependency Views module is active in all instances, and includes demo data.

Dependency Views sample map



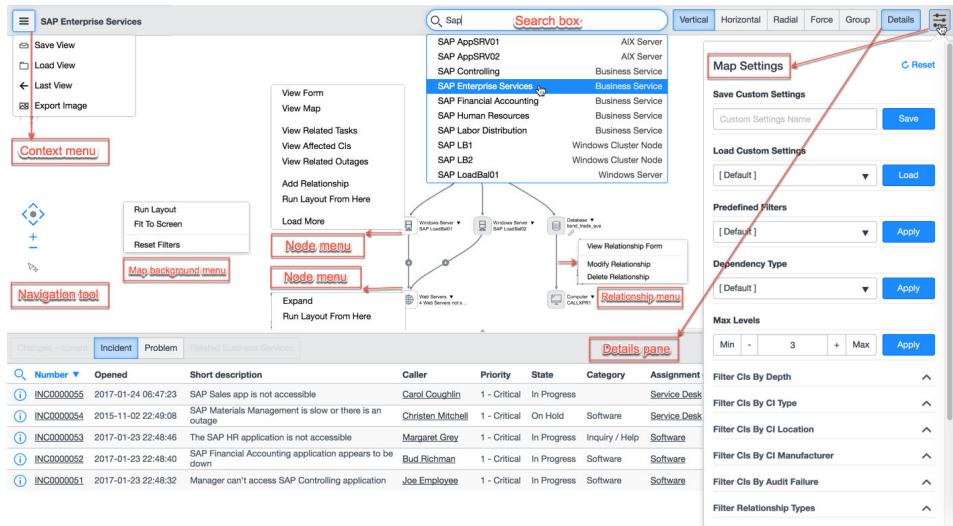
When you click the map icon [] on a CI record or on a task record that identify a CI, a map opens.

Roles

Users with the itil and ecmdb_admin roles can view maps and perform all actions in the map. Actions include access to the map views and saved filters, both from the lists in the map and from the **Saved Filters** module.

Dependency Views map menus and controls

Dependency Views maps contain the following menus and controls.



Map options

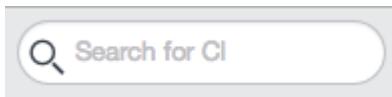
The following options are available across the top of the map.



Menu to save, load and export views of the map.

<Root CI>

Next to the menu icon is the name of the current root node (CI) of the map.



Enter the name of a CI, application service, or business service to load into the map. Alternatively, you can start typing to have the auto-complete feature present a list of CIs and services that match your partial value.

Vertical	Display the map in vertical view.
Horizontal	Display the map in horizontal view.
Radial	Display the map in radial view.
Force	Centers the elements around the parent CI, regardless of upstream or downstream relationships.
Group	Groups the elements according to their CI type. Displays related lists such as Problems, Changes and Related Services that are associated with the selected CI. <ul style="list-style-type: none">Click a service to highlight the CIs that are associated with that service.Click Related Services, then double-click a service to display the map in the Event Management dashboard. If the Event Management plugin is active, then events and alerts are also displayed.
Details	 Set filters for the map. Use the navigation tools to increase or decrease the view of the map, rearrange the icons on the map, and move the map on the page. <ul style="list-style-type: none">Use the plus sign (+) to increase magnification of the map.

- Use the minus sign (-) to decrease magnification of the map.
- Click the center dot to center the map on the page.
- Use the direction arrows to move the page in that direction.
- Use the selection tool under the navigation tool to toggle between moving the entire map or moving one CI on the map.

Map menu

The following options are available if you right-click the map background.

Run Layout	Redraws the map with the current layout option.
Fit To Screen	Resizes the map to fit all the nodes in the map window.
Reset Filters	Performs the same action as the Filters > Reset option.

Node menu

The following options are available if you right-click a node.

View Form	Displays the CMDB record of the selected CI in a new tab of the browser.
View Map	Reloads the map using the selected CI as the new root node, with the currently defined layout

	setting. This option does not display on the root node.
View Related Tasks	Displays all tasks or outages associated with the selected CI, including incidents, problems, change requests, and follow-on tasks. This option is always available, even if there are no tasks associated with the CI. This option does not appear on collapsed nodes.
View Affected CIs	Shows a list of all tasks that have the CI listed as an Affected CI. This option is only visible when you access the map from the map icon in a task record's Configuration item field.
View Related Outages	Displays all outages involving the selected CI. This option only appears when there is an outage associated with the CI. This option does not appear on collapsed nodes.
Add Relationship	This option displays a dotted green line that you can drag to another CI to create a relationship link. A popup dialog allows you to define the relationship type.
Expand	Displays all CIs and components within a clustered node, or virtual groups (virtual nodes that appear when <code>glide.bsm.too_many_children</code> is reached). This option appears only if the node is a cluster node or a virtual group node.

If **Load More** was previously used, then **Expand** reverts the results of the **Load More** operation.

The number of additional icons to display is bound by the value of the glide.bsm.max_nodes property.

Collapse

Collapses all CIs and components within a cluster node back to a single node. Also, collapses a virtual group that has been expanded. This option only appears if the node has been expanded using the **Expand** menu item.

If **Load More** was previously used, then **Expand** reverts the results of the **Load More** operation.

Run Layout From Here

This option re-runs the chosen layout using the current node. Use this option to get a new or clearer view on the same map.

Load More

Starting at the selected icon, loads the next level of the map, past the setting of **Max Levels**.

Virtual grouping is not applied at the newly loaded level even if the criteria for virtual grouping is met.

The number of additional icons to display is bound by the value of the glide.bsm.max_nodes property.

Relationship menu

The following options are available if you right-click a relationship link.

View Relationship Form	Opens the CI Relationship form. You can modify the Parent , Type , and Child of the relationship from this form.
Modify Relationship	Searches for and selects a new relationship for this link.
Delete Relationship	Deletes a relationship. The relationship is deleted after prompting for confirmation.

Cluster nodes in a Dependency Views map

Dependency Views maps can display cluster group nodes alongside individual CI nodes, and the child nodes of these cluster groups.

Clusters are CIs in the Cluster [cmdb_ci_cluster] table. A cluster CI is an organized set of computer CIs that work together as a single system. Each node in a cluster group represents a CI, typically a server, that can have referenced hardware, such as disks and network adapters.

Cluster nodes on a Dependency Views map can display in two modes:

- Collapsed mode: Displays only the cluster CI node without its child CI nodes. This mode avoids unnecessary clutter in large maps.
- Expanded mode: Displays the cluster CI node and all its child CI nodes.

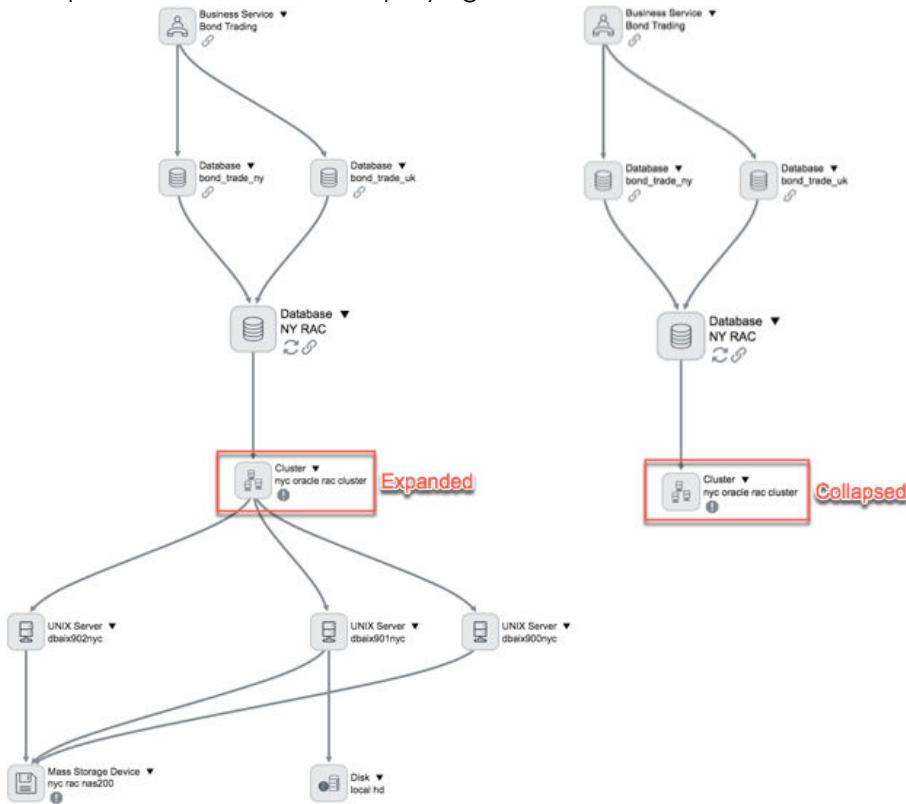
Menu options available for a clustered node include **Collapse** and **Expand**, which allow you to control the density on the map.

By default, Dependency Views collapses all cluster groups and displays clusters in collapsed mode on the map.

Annotation

Icons for cluster nodes and cluster group CI nodes are noted by the string "Cluster" and by a unique cluster icon. The system searches through all the component nodes in a cluster CI or collapsed node looking for tasks, outages, and trouble, such as incidents, problems, or change requests. This search evaluates only the number of levels that are displayed in the diagram.

An expanded cluster node displaying its child nodes



Virtual grouping of nodes in a Dependency Views map

To reduce the density on a map, Dependency Views automatically groups CIs of a similar CI type from the same level.

A large number of nodes can cause a Dependency Views map to become too dense to be helpful. Therefore, if the number of nodes with a similar CI type from same level, exceeds the value of the Maximum number of nodes (of a similar CI type and at the same level) to display before applying virtual grouping property, then those nodes are automatically grouped into a virtual group. A single node, the virtual group node is displayed to represent the virtual group, while all actual nodes in the virtual group (that are of a similar CI type), are hidden. Virtual group nodes represent CIs of a similar CI type but are not CIs by themselves and cannot have tasks assigned to them. The number of actual collapsed nodes in the virtual group is noted on the virtual group node.

By default, child nodes of a virtual group are not displayed. You can enable the Show children of virtual groups property to display child nodes underneath virtual groups.

Virtual grouping is not applied at the level underneath a virtual group even if the criteria for virtual grouping is met (the number of nodes with a similar CI type from that level exceeds the preconfigured property value). However, virtual grouping can happen at the following level if that criteria is met. This behavior does not depend on any property settings, and you cannot change it.

Menu options for a virtual group include **Expand** and **Collapse**, which allow you to apply virtual grouping and display only the virtual group node, or to undo the virtual grouping and display all actual nodes.

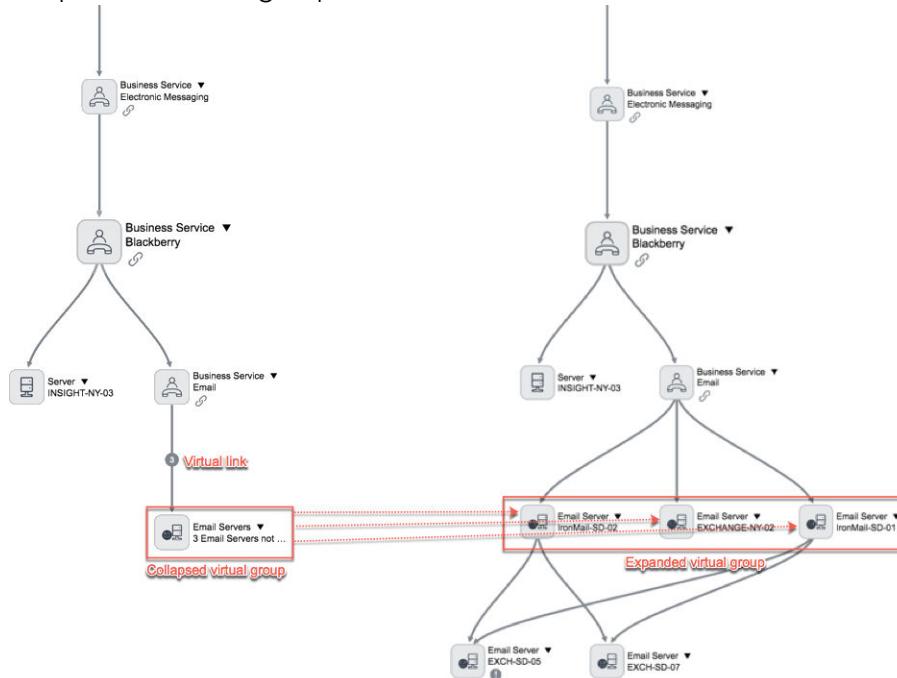
Virtual links

A virtual node is connected to other nodes with a virtual link. A virtual link denotes that there such link between at least one CI in the virtual group, to another CI node on the map.

Note: Predefined filters do not apply to virtual groups. Therefore a virtual group displays even if it contains CIs that a predefined filter would have excluded. Upon the expansion of a virtual group, predefined filters are applied, and any or all of the CIs that were previously virtually grouped, might no longer display on the map.

Also, when using the node menu option **Load More**, virtual grouping is not applied at the newly loaded level even the criteria for virtual grouping is met.

An expanded virtual group



Related reference

- Properties for Dependency Views

Use Dependency Views

Use the layout controls on a Dependency Views map to display elements in different configurations for easier management. Use the filter panel on

the map to display fewer levels or to filter out elements you don't want to see, then save the filter for use later. Draw new relationships between elements or edit existing relationships.

- [View a Dependency Views map](#)

When you display a Dependency Views map using one of the options below, the map is centered on the root CI, and displays the layout and number of levels defined in the map properties. If Operational Intelligence is activated, then a Dependency Views map provides a mode that lets you directly access metrics information for the CIs on the map.

- [Save or load a Dependency Views map](#)

In the **View Map** module, use the menu icon to save and load Dependency Views maps.

- [Delete a saved Dependency Views map view](#)

Use the **Saved Views** module to delete a previously saved view.

- [Change the layout of Dependency Views map](#)

You can select from different layout options for your Dependency Views map.

- [Filter the view of a Dependency Views map](#)

Filter a Dependency Views map to display specific types or categories of configuration items.

- [View metrics for CIs in a Dependency Views map](#)

Operational Intelligence processes metrics data for CIs, calculates statistics and aggregations, and detects metrics anomalies. A Dependency Views map lets you switch to metrics mode to directly access the Insights Explorer that displays metrics data for CIs on the map.

- [Perform actions on nodes in a Dependency Views map](#)

You can view various related items for the nodes in a Dependency Views map.

- [Export a Dependency Views map](#)

Export a Dependency Views map to an image in PNG format.

- [View collapsed nodes in a Dependency Views map](#)

Cluster and virtually grouped nodes can be displayed in a collapsed mode to avoid unnecessary clutter in large maps.

Related concepts

- [Domain separation and Dependency Views](#)
- [Dependency Views map](#)
- [Cluster nodes in a Dependency Views map](#)
- [Virtual grouping of nodes in a Dependency Views map](#)

Related reference

- [Supported browsers for Dependency Views](#)
- [Dependency Views map menus and controls](#)
- [Properties for Dependency Views](#)
- [Components installed with Dependency Views](#)

Related topics

- [Administer Dependency Views](#)

When you display a Dependency Views map using one of the options below, the map is centered on the root CI, and displays the layout and number of levels defined in the map properties. If Operational Intelligence is activated, then a Dependency Views map provides a mode that lets you directly access metrics information for the CIs on the map.

Before you begin

General role requirements:

- To access a Dependency Views map from either the navigation menu, a script API, or directly from a URL, the minimum role required is the dependency_views. Some operations that are related to icons, indicators, and menu actions require the ecmdb_admin role. Some operations that are related to properties and dependency types require the admin role.
- Dependency Views enforces ACL permissions on CIs, and visually hides them and their relationship from the map if the permission requirement is not met.

About this task

The maps generated by Dependency Views are based on D3 and Angular technology, providing a modern interactive graphical interface to visualize configuration items and their relationships.

If Service Mapping is activated, Dependency Views maps are enhanced to display dependencies that reflect connections in service maps. In addition, the list of related services in the **Details** section, includes application services, and technical and manual services if Event Management is activated. All CIs that are included in a service, are displayed underneath the service node on the map.

Maps provided by Service Mapping are for application services, including comprehensive maps from the perspective of application services. For more information, see [Service Mapping](#).

Administrators can configure the setting for the default layout of the map and number of levels displayed. When you access the map from a saved view, the map opens using the properties in the saved view, and not the default map properties.

Procedure

Navigate to **All > Dependency Views** and open one of these modules:

- **View Map in New Tab:** Opens the map in a new, full screen tab without the application navigator.
- **View Map:** Opens the map in the content pane of the current tab.
- **Saved Views:** Opens a view of a map that you previously saved.

Click a number in the **Version** column, and then click the () icon.

Related tasks

- Save or load a Dependency Views map
- Delete a saved Dependency Views map view
- Change the layout of Dependency Views map
- Filter the view of a Dependency Views map
- View metrics for Cls in a Dependency Views map
- Perform actions on nodes in a Dependency Views map
- Export a Dependency Views map
- [View collapsed nodes in a Dependency Views map](#)

In the **View Map** module, use the menu icon to save and load Dependency Views maps.

1. Navigate to **All > Dependency Views > View Map**.

2. Click the view menu icon ().

3. Select **Save View**, **Load View**, or **Last View**.

Related tasks

- View a Dependency Views map
- Delete a saved Dependency Views map view
- Change the layout of Dependency Views map
- Filter the view of a Dependency Views map
- View metrics for Cls in a Dependency Views map
- Perform actions on nodes in a Dependency Views map
- Export a Dependency Views map

- View collapsed nodes in a Dependency Views map

Use the **Saved Views** module to delete a previously saved view.

Before you begin

Role required: admin

Procedure

1. Navigate to **All > Dependency Views > Saved Views**.
2. Use the checkbox in the first column of the table to select the map view that you wish to delete.
3. Select **Delete** from the **Actions on selected rows** drop-down menu.

Related tasks

- View a Dependency Views map
- Save or load a Dependency Views map
- Change the layout of Dependency Views map
- Filter the view of a Dependency Views map
- View metrics for CIs in a Dependency Views map
- Perform actions on nodes in a Dependency Views map
- Export a Dependency Views map
- View collapsed nodes in a Dependency Views map

You can select from different layout options for your Dependency Views map.

Before you begin

Role required: none

Procedure

1. Navigate to **All > Dependency Views > View Map**.
2. Select one of the following layout options from the menu across the top of the view.
 - **Vertical:** Displays the elements in a vertical tree pattern according to their upstream and downstream relationships. This is the default value for the initial display of the map.
 - **Horizontal:** Displays the elements in a horizontal tree pattern according to their upstream and downstream relationships.
 - **Radial:** Displays the elements in a radial pattern according to their upstream and downstream relationships.
 - **Force:** Centers the elements around the parent CI, regardless of upstream or downstream relationships.
 - **Group:** Groups the elements according to their CI type.
 - **Details:** Displays related alerts, incidents, problems, and related services.

Related Services displays application services related to the CIs currently displaying in the map. If Event Management is activated then technical services and manual services are included. You can double-click a service to display the map in the Event Management dashboard.

Related tasks

- [View a Dependency Views map](#)
- [Save or load a Dependency Views map](#)
- [Delete a saved Dependency Views map view](#)
- [Filter the view of a Dependency Views map](#)
- [View metrics for CIs in a Dependency Views map](#)
- [Perform actions on nodes in a Dependency Views map](#)

- Export a Dependency Views map
- View collapsed nodes in a Dependency Views map

Filter a Dependency Views map to display specific types or categories of configuration items.

Before you begin

Role required: admin

About this task

Use the filter panel to control which elements of the map are displayed and to save versions of a filter for later use.

Procedure

1. Navigate to **All > Dependency Views > View Map**.



2. Click the button to open **Map Settings**.

Filter panel strips and options	Description
Save Custom Settings	Configure desired custom settings, then enter a name and click Save . Custom settings can be loaded by using the Load Saved Custom Settings option. Navigate to Dependency Views > Saved Settings > to display all saved custom settings.
Load Custom Settings	Apply previously saved custom settings to the current map.
Predefined Filters	Apply previously defined filters consisting of configuration type,

Filter panel strips and options	Description
	<p>CI type, and relationship filters. You can Set a predefined filter as default.</p> <p>This filter is applied first, before any other filters (such as Filter CIs by Depth) are applied.</p>
Dependency Type	Apply a filter that runs in real time and generates a custom view of a service map for a specific CI.
Max Levels	Designate how many levels from the root CI display on the map.
Filter CIs by Depth	Designate which levels of CI display on the map.
Filter CIs by CI Type	Designate what CI types display in the map.
Filter CIs By CI Location	Designate what CI locations display in the map.
Filter CIs By CI Manufacturer	Designate what CI manufacturers display in the map.
Filter CIs By Audit Failure	Hides CIs that failed the CMDB health staleness test. This option is available only if there are any such CIs.
Filter Relationship Types	Designate what relationship types display in the map.
Map Indicators	Designate what types of tasks display and get counted in the map.

Filter panel strips and options	Description
Remove Filtered Items	Off: Gray out filtered items on the map. On: Do not display filtered items on the map.
Run Layout Automatically	On: The configured layout to the map is reapplied whenever the filter is changed. Off: The map layout remains static when the filter is changed.
Fit to Screen Automatically	On: The map magnification will increase or decrease automatically to display all CIs on the map. Off: The map magnification remains unchanged when the map is reloaded.

3. Click a filter strip to expand or collapse it, and to set filter items.

Related tasks

- [View a Dependency Views map](#)
- [Save or load a Dependency Views map](#)
- [Delete a saved Dependency Views map view](#)
- [Change the layout of Dependency Views map](#)
- [View metrics for CIs in a Dependency Views map](#)
- [Perform actions on nodes in a Dependency Views map](#)
- [Export a Dependency Views map](#)

- View collapsed nodes in a Dependency Views map

Operational Intelligence processes metrics data for CIs, calculates statistics and aggregations, and detects metrics anomalies. A Dependency Views map lets you switch to metrics mode to directly access the Insights Explorer that displays metrics data for CIs on the map.

Before you begin

The Operational Intelligence (com.snc.sa.metric) plugin must be activated to enable this functionality, and metrics data needs to be processed for the CIs on the Dependency Views map.

Role required: admin

About this task

Open a Dependency Views map in metric mode which integrates a Dependency Views map with the Insights Explorer functionality that is tailored to the map. In this mode, you can access Insights Explorer functions directly from the map, to explore metrics data for the CIs on the map. All map CIs are accessible in the right hand side pane, from where you can drill into metrics data.

Procedure

1. Navigate to **All > Dependency Views > View Map** to open a map.
2. Right-click on a CI on the map and select **View Metrics** to open the Dependency View map in metrics mode.
In the panel on the right side, the CI that you selected on the Dependency View map is selected by default, and the list of all the metrics available for that CI are displayed.
3. Click the '<' sign on the left of the CI to display all the CIs that you can explore metrics for.
The Insights Explorer is scoped for exploring only the CIs that currently display on the Dependency Views map, and you cannot add or remove CIs from the list. If you use map settings or filters to filter out CIs from the map, the same filtering will apply to the list of CIs that you can explore metrics for.
4. Click on a CI in the CIs list or right-click on a CI on the Dependency Views map, to drill down to the CI's metrics.

5. Click the **Dependencies Map** tab or the **Metrics** tab to switch modes:

- a. In **Metrics** mode: The full functionality of the Insights Explorer is available, you can create metric charts by dragging metrics into the canvas. You can modify chart settings, select different time ranges for the charts, and perform other actions as described in [View metric values in the Insights Explorer](#).
- b. In **Dependencies Map** mode: Select a CI on the map to drill down to its metrics data, drop-down the **Layout** list to choose a different layout, or modify map settings.

Related tasks

- [View a Dependency Views map](#)
- [Save or load a Dependency Views map](#)
- [Delete a saved Dependency Views map view](#)
- [Change the layout of Dependency Views map](#)
- [Filter the view of a Dependency Views map](#)
- [Perform actions on nodes in a Dependency Views map](#)
- [Export a Dependency Views map](#)
- [View collapsed nodes in a Dependency Views map](#)

You can view various related items for the nodes in a Dependency Views map.

About this task

If the node is a collapsed node or represents a cluster, the incidents, problems and change requests are for all the collapsed nodes.

Procedure

1. Navigate to **All > Dependency Views > View Map**.

2. Click the ▼ icon next to a node or right-click a node on the map, to access the following menu items:

Node Menu

View Form	Displays the CMDB record of the selected CI in a new tab of the browser.
View Map	Reloads the view using the selected CI as the new root node, with the currently defined layout setting. This option does not display on the root node.
View Related Tasks	Displays all tasks or outages associated with the selected CI, including incidents, problems, change requests, and follow-on tasks. This option is always available, even if there are no tasks associated with the CI. This option does not appear on collapsed nodes.
View Affected Cls	Shows a list of all tasks that have the CI listed as an Affected CI. This option is only visible when you access the view from the view icon in a task record's Configuration item field.
View Related Outages	Displays all outages involving the selected CI. This option only appears when there is an outage associated with the CI. This option does not appear on collapsed nodes.
Add Relationship	This option displays a dotted green line that you can drag to another CI to create a relationship link. A popup dialog

	allows you to define the relationship type.
Expand	Displays all Cls and components within a cluster node or a collapsed node. This option only appears if the node is a collapsed or cluster node.
Collapse	Collapses all Cls and components within a cluster node or a collapsed node back to a single node. This option only appears if the node has been expanded using the Expand menu item.
Run Layout From Here	This option re-runs the chosen layout using the current node. Use this option to get a new or clearer view on the same map.
Load More	Starting at the selected icon, loads the next level of the map, past the setting of Max Levels .

Related tasks

- [View a Dependency Views map](#)
- [Save or load a Dependency Views map](#)

- Delete a saved Dependency Views map view
- Change the layout of Dependency Views map
- Filter the view of a Dependency Views map
- View metrics for Cls in a Dependency Views map
- Export a Dependency Views map
- View collapsed nodes in a Dependency Views map

Export a Dependency Views map to an image in PNG format.

Before you begin

Role required: admin

Procedure

1. Navigate to **All > Dependency Views > View Map**.
2. Configure the map view as you want the image to appear.
The exported image displays the current view of the map.
3. Click the view menu icon ().
4. Click **Export Image**.
5. Right-click the image and select **Save Image As**, **Print**, or any other menu option.

Note: You can't export images from a Dependency Views map using Internet Explorer as your browser.

6. Click the "X" button to close the **Export Image** window.

Related tasks

- View a Dependency Views map
- Save or load a Dependency Views map
- Delete a saved Dependency Views map view

- Change the layout of Dependency Views map
- Filter the view of a Dependency Views map
- View metrics for CIs in a Dependency Views map
- Perform actions on nodes in a Dependency Views map
- View collapsed nodes in a Dependency Views map

Cluster and virtually grouped nodes can be displayed in a collapsed mode to avoid unnecessary clutter in large maps.

1. To expand a collapsed node, right-click the CI and select **Expand** from the context menu.
2. To collapse an expanded cluster node with children, right-click the CI and select **Collapse** from the context menu.

Related tasks

- View a Dependency Views map
- Save or load a Dependency Views map
- Delete a saved Dependency Views map view
- Change the layout of Dependency Views map
- Filter the view of a Dependency Views map
- View metrics for CIs in a Dependency Views map
- Perform actions on nodes in a Dependency Views map
- Export a Dependency Views map

Administer Dependency Views

Users with the admin role can control the appearance and behavior of Dependency Views by configuring map indicators, map related items, map icons, and menu actions.

- Create or modify map indicators

Dependency Views maps and application service maps, use icons to display additional information for a CI by displaying its related records such as alerts, outages, incidents and problems. These icons are called map indicator.

- [Create or modify map icons](#)

Upload new icons or modify existing icons to customize the icon displayed for a CI in maps in Dependency Views, Service Mapping, and Event Management.

- [Create a predefined filter](#)

Create filters to narrow down the CIs that are displayed on a Dependency Views map. You can create filters that are based on CIs' class, CIs' attributes, or CIs' relationships.

- [Set a predefined filter as default](#)

You can set a custom predefined filter as the default predefined filter for viewing maps.

- [Create or modify Map Related Items](#)

The Map Related Items module relates referenced CIs to one another, which allows them to be displayed in a Dependency Views map.

- [Create or modify Dependency Views menu actions](#)

To modify an existing menu option, first you create a copy of the original menu action record, and then you modify the copy.

- [Condition and script parameters for menu actions](#)

You can use the following condition and script parameters for menu actions.

- [Create or edit a dependency type](#)

Use one of the dependency types provided, or create a custom dependency type with a script that will execute in real time to generate a custom view of a Dependency Views map for a specific CI.

Related concepts

- [Domain separation and Dependency Views](#)
- [Dependency Views map](#)
- [Cluster nodes in a Dependency Views map](#)
- [Virtual grouping of nodes in a Dependency Views map](#)

Related reference

- [Supported browsers for Dependency Views](#)
- [Dependency Views map menus and controls](#)
- [Properties for Dependency Views](#)
- [Components installed with Dependency Views](#)

Related topics

- [Use Dependency Views](#)

Dependency Views maps and application service maps, use icons to display additional information for a CI by displaying its related records such as alerts, outages, incidents and problems. These icons are called map indicator.

Before you begin

Role required: admin

About this task

The default configuration includes map indicators for the following record types:

- Open incident.
- Open alert.
- Unplanned current outage.

- Planned current outage, or an open problem.
- Current, planned, or recent change request.

You can filter out the display of affected CIs, alerts, current change requests, incidents and problems from the map settings menu.

The Affected CI's map indicator appears for CIs in two related but not identical situations. It appears for CIs for which tasks such as change request, incident, or problem were directly created for, and for any CIs that were added in those tasks (parent tasks) as Affected CIs (The CI for which a task is directly created for, is automatically added as an affected CI in that task). The state of affected CI's depends on the status of the respective parent task. For as long as the parent task is active, the associated affected CIs continue to be impacted by the task issue. In a map, the Affected CI's indicator displays for all affected CIs for as long as the parent task is active. On a map, the Affected CI tooltip displays the details of the task records in which the CI was added as an affected CI. However, the **Details** pane does not contain an Affected CI's tab, and no further details about affected CIs, or the associated tasks are displayed. After the parent task is closed, the Affected CI's indicator no longer displays for any of the tasks' affected CIs. For information about affected CIs in Change Management, see [Associate CIs to a change request](#).

Note: Details about affected CIs are derived from the task and the cmdb_ci tables and their extensions. Therefore, if you use custom tables to store CIs for incidents, problems and changes, it affects the details that are displayed for affected CIs.

For more information on how map indicators are used to show tasks and outages in clusters and collapsed nodes, see [Cluster nodes in a Dependency Views map](#).

Procedure

1. Navigate to **All > Dependency Views > Map Indicators**.
2. Click **New** to create a new map indicator, or click the name of an indicator from the **Table** column to modify an existing map indicator.
3. Fill in the fields on the form, as appropriate.

Map Indicator form

Field	Description
Table	Name of the table represented by this map indicator. Note: The list shows only tables and database views that are in the same scope as the map indicator. Views are not supported, although included in the list.
Name	Name of the indicator.
Order	Priority order of the task. The highest priority task is the indicator with the lowest order number.
Icon	File name and path of the icon image file, which can be a system image. <ul style="list-style-type: none">• To create a new icon, see Create or modify map icons• To create or use a system image see Storing images in the database.
CMDB CI field	Name of the field on the selected table that contains the configuration item.
Start field	Record property that determines the time-point on the metric chart timeline for placing records in the Insights Explorer.

Field	Description
	Possible values depend on the selected Table . For example, the incident indicator has values such as Actual end , Actual Start , and Approval Set .
Description field	Name of the field on the selected table that contains the description of the configuration item.
Description	Text to display when hovering over the indicator. Alphanumeric characters and spaces are valid for this field.
Conditions	Condition builder that specifies for which CIs to apply this indicator. For example, a CI that has a current past outage is highlighted for 5 days. You can configure a condition to designate a different timeframe for what is considered to be current.
Active in Service Map	Toggle that you can enable to make the specified table available in the Settings dialog box for application service maps. You can then toggle between displaying or not displaying the respective records on the map.
Active Dependencies	Toggle that you can enable to make the specified table available in the Settings dialog box for Dependency Views maps. You can then toggle

Field	Description
	between displaying or not displaying the respective records on the map.
Active in Metrics	Enable to make the toggle for the specified table available in the Settings dialog box for the Insights Explorer. You can then toggle between displaying or not displaying the respective records on the Insights Explorer.
Label	Text to display for the indicator on the map.
Tooltip Label	Prefix portion of the tooltip (Tooltip Label : Tooltip info).
Tooltip Info	Suffix portion of the tooltip (Tooltip Label : Tooltip info).

4. Click **Submit** to enter a new map indicator or click **Update** to modify an existing map indicator.

Result

For an indicator to appear in a Dependency Views map, a CI must meet all filter conditions, and **Active Dependencies** must be selected.

Related tasks

- [Create or modify map icons](#)
- [Create a predefined filter](#)
- [Set a predefined filter as default](#)
- [Create or modify Map Related Items](#)
- [Create or modify Dependency Views menu actions](#)

- Create or edit a dependency type

Related reference

- Condition and script parameters for menu actions

Upload new icons or modify existing icons to customize the icon displayed for a CI in maps in Dependency Views, Service Mapping, and Event Management.

Before you begin

Role required: admin

About this task

The icons used in Dependency Views maps are listed in the Map Icons module. Records in the **Map Icons** list are arranged by CI classes, such as cmdb_ci_linux_server. The path to the default image files is <https://<instance name>.service-now.com/images/app.ngbsm/<image name.svg>>. For information about uploading images to the database, see [Storing images in the database](#).

Role required: admin or cmdb_admin roles are required to access the records in this table [ngbsm_icon] to upload new icons.

Procedure

- Navigate to **All > Configuration > CI Class Manager**, and do the following actions:
 1. Click **Hierarchy** to display the CI Classes list.
 2. Select a class to modify the icon for.
 3. In the class navigator bar, expand **Class Info** and then select **Basic Info**.
 4. On the Basic Info form, click **Icon**.
 5. In the Icons dialog box, select an icon and then click **Update**.
 6. On the **Basic Info** form, click **Update**.

- Navigate to **All > Dependency Views > Map Icons**, and do the following actions:
 1. Click **New** to create a new map icon or click the name of an existing icon in the **Label** column to modify an existing icon.
 2. On the form, fill in the fields.

Map Icons form

Field	Description
CI Type	Label or the informal name of the CI table that this icon represents in the view.
Icon	Name of the icon.
URL	Path to the icon image using the following format: /image name.svg Click the lock icon to enter a new path.

3. Fill in the fields on the form, as appropriate.
4. Click **Submit** to enter a new icon or click **Update** to modify an existing icon.

What to do next

You can modify a Dependency Views map indicator to use the new icon.

Related tasks

- Create or modify map indicators
- Create a predefined filter
- Set a predefined filter as default
- Create or modify Map Related Items
- Create or modify Dependency Views menu actions

- Create or edit a dependency type

Related reference

- Condition and script parameters for menu actions

Create filters to narrow down the CIs that are displayed on a Dependency Views map. You can create filters that are based on CIs' class, CIs' attributes, or CIs' relationships.

Before you begin

Role required: ecmdb_admin

About this task

Create a predefined filter that you can then select to determine the scope of the CIs that are displayed in a Dependency Views map.

Configuration type filters filter by CI class, **CI filters** filter by CI attributes, and **relationship filters** filter by relationships. Only CIs that match at least one of the configuration type filters (if any exists), and at least one of the CI filters (if any exists), and at least one of the relationship type filters (if any exists) - are displayed on the map. If no filters are defined, then no filtering is applied.

Note: Predefined filters do not apply to virtual groups. Therefore a virtual group displays even if it contains CIs that a predefined filter would have not included. Upon the expansion of a virtual group, predefined filters are applied, and any or all of the CIs that were previously virtually grouped, might no longer display on the map.

Procedure

1. Navigate to **All > Dependency Views > Predefined Filters**.
2. On the **Predefined Filters** page, click **New**.
3. Type in a **Name** for the filter.
4. Click **Roles**, and in the **Roles** dialog box, select the roles that this filter will be available for.

5. Right-click on the page header, and click **Save**.
6. Create a configuration type filter:
 - a. Click **Configuration Types**, and then click **Edit**.
 - b. In the **Collection** slushbucket, select the classes that CIs must belong to in order to be displayed on the map, and move them to the **Configuration Types List**.
 - c. Click **Save**.
7. Create a CI filter:
 - a. Click **CI Filters**, and then click **New**.
 - b. In the **CI Filters** page enter conditions to filter CIs by specific attribute values.
 - c. Click **Submit**.
8. Create a relationship type filter:
 - a. Click **Relationship Type**, and then click **New**.
 - b. In the **Collection** slushbucket, select the relationships that CIs must have in order to be displayed on the map, and move them to the **Relationship Types List**.
 - c. Click **Save**.

What to do next

After creating a predefined filter, you can apply it to a map:



1. Click the icon to open **Map Settings**.
2. Select a filter from the **Predefined Filters** list.
3. Click **Apply**.

Related tasks

- [Create or modify map indicators](#)

- Create or modify map icons
- Set a predefined filter as default
- Create or modify Map Related Items
- Create or modify Dependency Views menu actions
- Create or edit a dependency type
- Filter the view of a Dependency Views map

Related reference

- [Condition and script parameters for menu actions](#)

You can set a custom predefined filter as the default predefined filter for viewing maps.

Before you begin

Role required: admin

Procedure

1. Create the custom predefined filter to be used as the default predefined filter.
2. On the predefined filter form, click the context menu and select **Copy sys_id**.
3. Navigate to **User Administration > User Preferences**.
4. Click **New** and create a new user preference record using these values:
 - **Name:** ecmdb.ciview
 - **Type:** String
 - **Value:** Paste the sys_id of the custom predefined filter
 - **User:** Leave blank to create a system-wide setting
 - **Description:** Description of the predefined filter

- **System:** Selected

5. Click **Submit**.

What to do next

In **Map Settings**, when you select the **Default** option for **Predefined Filters**, the custom predefined filter that was set, will be applied.

Related tasks

- Create or modify map indicators
- Create or modify map icons
- Create a predefined filter
- Create or modify Map Related Items
- Create or modify Dependency Views menu actions
- Create or edit a dependency type
- Filter the view of a Dependency Views map

Related reference

- Condition and script parameters for menu actions

The Map Related Items module relates referenced CIs to one another, which allows them to be displayed in a Dependency Views map.

Before you begin

Role required: admin

About this task

The base system configuration includes the following tables and relates them to items in the Computer [cmdb_ci_computer] and Server [cmdb_ci_server] tables.

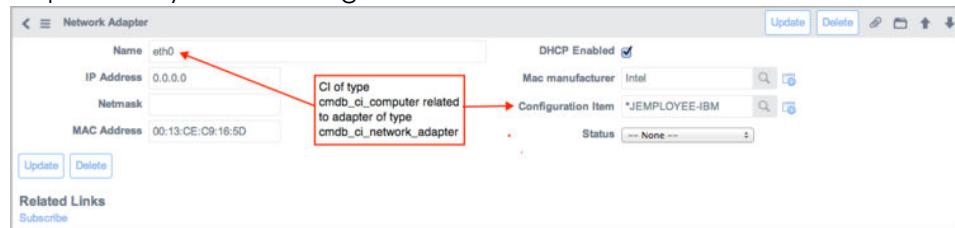
- Disk [cmdb_ci_disk]

- Network Adapter [cmdb_ci_network_adapter]
- Database [cmdb_ci_database]

Some additional referenced CIs that can be related in this manner are file systems and running processes.

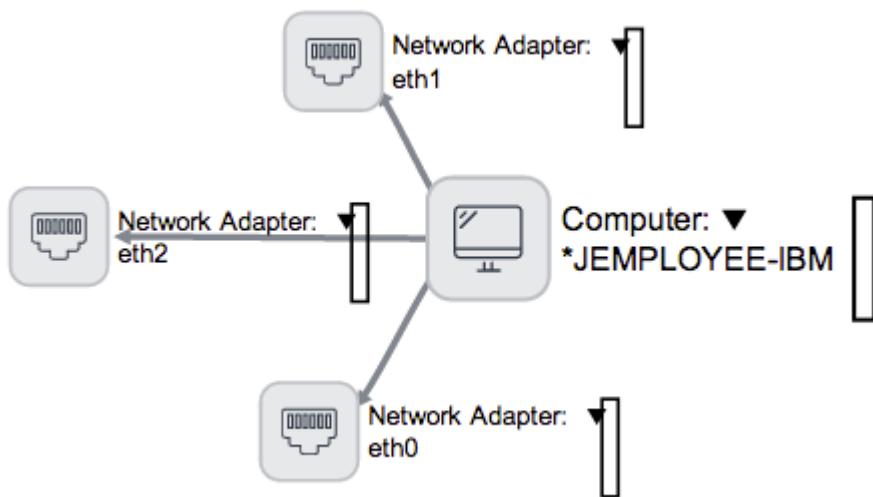
In the following example, computer nodes in the map are related to network adapter nodes if the **Configuration Item** field of the adapter records reference the specific CI node. Access or create a network adapter record from the Network Adapter related list in the cmdb_ci_computer record.

Dependency Views Configuration Item field



The Dependency Views map for the *JEMPLOYEE-IBM computer shows the network adapter attached to the computer.

Dependency Views map Related Items example



You can configure Dependency Views to display CIs that have no relationship record, but are related to other CIs by reference fields.

Procedure

1. Navigate to **All > Dependency Views > Map Related Items**.
2. Click **New** to create a new related item, or click in the row of an existing CI to modify an existing map related item.
3. On the form, fill in the fields.
See the Related Items form table.
4. Click **Submit** to enter a new map related item or click **Update** to modify an existing map related item.

Related Items form

Control	Description
Configuration item	CI that represents the base node or a CI in a table that extends the base node table. In the base system, the configuration item that represents the base node is Computer [cmdb_ci_computer], which includes all types of workstations and servers.
Related item	Table name of the related item. Only the cmdb_ci table and tables that extend it are displayed in the choice list.
Related field	Field that links this related item to the configuration item. In many cases, the appropriate value is automatically populated in the field after the first two fields are selected. Select the drop-down menu for additional options.
Active	Check box to enable or disable this record.

Related tasks

- Create or modify map indicators
- Create or modify map icons
- Create a predefined filter
- Set a predefined filter as default
- Create or modify Dependency Views menu actions
- Create or edit a dependency type

Related reference

- Condition and script parameters for menu actions

To modify an existing menu option, first you create a copy of the original menu action record, and then you modify the copy.

Before you begin

Role required: admin

About this task

This ensures that your instance can update the record normally during the upgrade process and allows you to quickly restore the original menu option, if necessary.

Procedure

1. To create a new menu option, navigate to **Dependency Views > Map Menu Actions** and click **New**.
Fill in the fields on the form, as appropriate. See the Menu Action form table.
2. To modify an existing menu option, navigate to **Dependency Views > Map Menu Actions**.
3. Open the menu action you want to edit.

4. Right-click in the header and click **Insert and Stay**.
This step creates a duplicate copy of the menu action and leaves it open for editing.
5. Change the name of the copied record to avoid confusion.
6. Modify the form fields as necessary and save the record.
7. Open the original record and disable it by clearing the **Active** check box.

Menu Action form

Control	Description
Name	Descriptive name that appears as the menu option.
Active	Check box that allows you to enable or disable this record.
Condition	Condition that triggers the display of this menu option. If the condition evaluates to false the menu option does not display. Script is evaluated in JavaScript in the user's browser and does not have access to all the APIs that Business Rules do. For details on available parameters, see Condition Parameters..
Item	Map element for which the menu option displays. Valid values are: <ul style="list-style-type: none">• Canvas for the menu on the map background.• Node for the menu on a CI.• Relationship for the menu on a relationship link.

Control	Description
Order	Physical location of the option in the menu. The option with the lowest order number appears first in the menu. All editable and custom options appear below the permanent menu options.
Script	Script that is executed in the browser when the menu option is selected. Script is evaluated in JavaScript in the user's browser and does not have access to all the APIs that Business Rules do.
Type	Menu action type being created, either a menu option or a menu separator. The menu separator is a single line. When the type is a separator, the Script field is ignored.

Related tasks

- [Create or modify map indicators](#)
- [Create or modify map icons](#)
- [Create a predefined filter](#)
- [Set a predefined filter as default](#)
- [Create or modify Map Related Items](#)
- [Create or edit a dependency type](#)

Related reference

- Condition and script parameters for menu actions

You can use the following condition and script parameters for menu actions.

Condition parameters

Note: The usual regular expression conventions are valid in the condition field, such as ! for NOT, && for AND, and || for OR.

The **Condition** field contains a boolean expression that evaluates to true or false. If the condition is true or if there is no condition, the specified option appears in the menu when you right-click a CI or a relationship link. When you select the option from the menu, ServiceNow executes the associated script.

Common Elements for Building a Condition

Text	Description
item	Node or reference link's data on which you performed the right-click action.
item.label	Label of the node.
item.ci_type	CI's type (table), such as <code>cmdb_ci_service</code> .
item.name	Name of CIs. CI's type name or the table label.
item.location	Location of the CI, such as New York.
item.manufacturer_name	Name of the CI's manufacturer, such as Dell Inc.
item.id	The <code>sys_id</code> of the CI.

Text	Description
item.is_selected	The item that is selected in the map.
item.level	The current default level.
item.locationId	The sys_id of the CI node's location.
item.locationName	The full address of the location.
item.manufacturerId	The sys_id of the CI's manufacturer.

Valid Conditions for Condition Parameters

Condition	Description
item.is_collapsed	The node is a collapsed node.
item.is_cluster	The node is a cluster node.

Script parameters

Menu action scripts are executed on the client when a user clicks the menu option. You can use the same building blocks in scripts as in conditions. Menu action scripts do not function on separators. These are some additional, useful expressions for scripts:

Condition	Description
item.id	The sys_id of the CI node or relationship link.
item.source	The sys_id of the relationship's parent or child.
item.target	The sys_id of the relationship's parent or child.

Condition	Description
item.label	The name of the CI node, such as IronMail-SD-02.
item.location	The sys_id of the CI node's location.
item.location_name	The full address of the location, such as 4616 Clairemont Drive, North Clairemont, San Diego CA.
item.manufacturer_id	The sys_id of the CI's manufacturer.

Related tasks

- [Create or modify map indicators](#)
- [Create or modify map icons](#)
- [Create a predefined filter](#)
- [Set a predefined filter as default](#)
- [Create or modify Map Related Items](#)
- [Create or modify Dependency Views menu actions](#)
- [Create or edit a dependency type](#)

Use one of the dependency types provided, or create a custom dependency type with a script that will execute in real time to generate a custom view of a Dependency Views map for a specific CI.

Before you begin

Role required: none

About this task

Create a JavaScript to customize the map. The script must comply with JavaScript syntax guidelines and the directions in the default script

template, and it can call platform APIs. Use a dependency type, for example:

- To narrow down and simplify a map, leaving out CIs that are not important for a specific task.
- To include only specific CIs that are hidden by default, such as qualifiers, end-points, and entry points.
- To display virtual relationships that are calculated, and that otherwise do not exist in the CMDB.
- As a tool to plan a new topology deployment that is based on existing resources.

The following dependency types are available in the base system:

Default

The default setting in the base system. With this setting, there is no processing of the dependency map through any dependency type scripts that might filter or modify the map.

Show All Relationships

Returns all qualifiers, end points, and entry points. This dependency type is available in the base system and is disabled by default. Typically, you would enable this dependency type for debugging and tracking purposes.

The following dependency types are available with [Service Mapping](#):

Application to Network Devices

Returns the network devices in the network paths leading to/from the given CI.

Network Device to Applications

Returns the applicative CIs which are target or source of network paths containing the given network device. In addition, returns the hosts of those applicative CIs, and for an applicative CI that is an inclusion, its parent CI is returned too.

Physical Network Connections

Returns hosts/network devices that are physically connected to the given host or network device.

Flow Dependencies

Returns all the server to server connections that were discovered using the Netflow collector. The script builds a graph based on data in the [sa_flow_server_comm] table. This table contains pairs of services represented by an IP and a listening port that are communicating with each other. For more information, see [Data collection and discovery using Netflow](#) and [Data collection and discovery using VPC Flow Logs](#).

Procedure

1. Navigate to **All > Dependency Views > Dependency Types**.
2. In the **Load Filter Scripts** list view, select an existing dependency type, or click **New**.
3. Enter or modify a script, adhering to the guidelines and requirements in the script template that is provided.
4. Click **Submit**.

Result

In a Dependency Views map, you can click **Dependency Type** to apply a custom script defined in a dependency type.

Related tasks

- [Create or modify map indicators](#)
- [Create or modify map icons](#)
- [Create a predefined filter](#)
- [Set a predefined filter as default](#)
- [Create or modify Map Related Items](#)
- [Create or modify Dependency Views menu actions](#)

Related reference

- Condition and script parameters for menu actions

Properties for Dependency Views

Use Dependency Views properties to configure how data appears in Dependency Views maps.

These properties are available for Dependency Views. To view and edit these properties, the admin role is required.

Properties for Dependency Views

Property	Description
Maximum number of CIs to display on a map at once. <code>glide.bsm.max_nodes</code>	The maximum number of nodes to retrieve from the database. If more nodes exist in the database, they are not displayed in the map. <ul style="list-style-type: none">Type: IntegerDefault value: 1000Location: Dependency Views > Map Properties
Maximum level depth from the root CI that can be initially displayed in Dependency Views. <code>glide.bsm.max_levels</code>	Level depth is the graph distance between the root CI and a node. <ul style="list-style-type: none">Type: IntegerDefault value: 3Other possible values: 1-49Location: Dependency Views > Map Properties

Property	Description
<p>Display the continuation of the map underneath virtual group. Virtual links are used to connect virtual groups to their child nodes.</p> <p>glide.bsm.show_virtual_node_children</p>	<ul style="list-style-type: none"> Type: Yes No Default value: No Location: Dependency Views > Map Properties
<p>Maximum number of child nodes to display (the rest will be collapsed).</p> <p>glide.bsm.too_many_children</p>	<p>Maximum number of nodes (of a similar CI type and at the same level) to display before applying virtual grouping.</p> <p>Nodes are collapsed for the map to meet this limit.</p> <ul style="list-style-type: none"> Type: Integer, valid values 1 or greater Default value: 10 Location: Dependency Views > Map Properties
<p>A value of true indicates that filtered out items will be removed from the graph along with any disconnected children while a value of false indicates that the items will be dimmed in color.</p> <p>glide.ngbsm.filters_remove_filtered_items</p>	<ul style="list-style-type: none"> Type: Yes No Default value: Yes Location: Dependency Views > Map Properties

Property	Description
<p>Maximum number of relations per node.</p> <p>glide.bsm.max_num_rels</p>	<p>The maximum number of relations to retrieve from the database. If more relations exist in the database, they are not displayed in the map.</p> <ul style="list-style-type: none"> • Type: Integer • Default value: 100 • Other values: 1 or greater • Location: Dependency Views > Map Properties
<p>A value of true indicates that when filters are changed the graph will recalculate its layout using the currently selected layout algorithm.</p> <p>glide.ngbsm.filters_run_layout Automatically</p>	<ul style="list-style-type: none"> • Type: Yes No • Default value: Yes • Location: Dependency Views > Map Properties
<p>A value of true indicates that when filters are changed the graph will be fit to the screen automatically.</p> <p>glide.ngbsm.filters_fit_to_screen_au tomatically</p>	<ul style="list-style-type: none"> • Type: Yes No • Default value: No • Location: Dependency Views > Map Properties
<p>A value of true allows relationship lines to be drawn using smooth curves instead of straight line segments. These curves can be</p>	<ul style="list-style-type: none"> • Type: Yes No • Default value: Yes

Property	Description
more taxing on the browser, setting to false may improve fluidity of animation and interaction for Dependency Views. glide.ngbsm.performance_allow_curves	<ul style="list-style-type: none"> Location: Dependency Views > Map Properties
Amount of time in milliseconds a notification stays on the screen. glide.ngbsm.notification_display_time	<ul style="list-style-type: none"> Type: Integer Default value: 5000 Location: Dependency Views > Map Properties
The maximum amount of results displayed when searching for CIs. glide.ngbsm.search_ci_limit	<ul style="list-style-type: none"> Type: Integer Default value: 10 Location: Dependency Views > Map Properties
The maximum amount of results displayed when searching for Relationship Types. glide.ngbsm.search_rel_type_limit	<ul style="list-style-type: none"> Type: Integer Default value: 5 Location: Dependency Views > Map Properties
When available, the map should display the class labels for each CI. glide.ngbsm.show_class_labels	<ul style="list-style-type: none"> Type: Yes No Default value: Yes

Property	Description
	<ul style="list-style-type: none"> Location: Dependency Views > Map Properties
Truncate node labels to a single line and to fit available space (default). Disable to display entire labels on multiple lines and wrapped as needed. glide.ngbsm.truncate_long_labels	<p>If glide.ngbsm.show_class_labels is enabled, then the class label always displays on top of the CI label, and wrapping applies to both the class and the CI labels.</p> <ul style="list-style-type: none"> Type: Yes No Default value: No Location: Dependency Views > Map Properties
Minimum horizontal distance between nodes in horizontal layout. glide.bsm.layout_horizontal_spacing_x	The distance is measured in pixels between one node's center to another node's center. <ul style="list-style-type: none"> Type: Integer Default value: 200 Location: Dependency Views > Map Properties
Minimum vertical distance between nodes in horizontal layout. glide.bsm.layout_horizontal_spacing_y	The distance is measured in pixels between one node's center to another node's center. <ul style="list-style-type: none"> Type: Integer Default value: 100

Property	Description
	<ul style="list-style-type: none">• Location: Dependency Views > Map Properties
Minimum horizontal distance between nodes in vertical layout. glide.bsm.layout_vertical_spacing_x	The distance is measured in pixels between one node's center to another node's center. <ul style="list-style-type: none">• Type: Integer• Default value: 125• Location: Dependency Views > Map Properties
Minimum vertical distance between nodes in vertical layout. glide.bsm.layout_vertical_spacing_y	The distance is measured in pixels between one node's center to another node's center. <ul style="list-style-type: none">• Type: Integer• Default value: 125• Location: Dependency Views > Map Properties

Components installed with Dependency Views

Several types of components are installed with the activation of the Next_Gen BSM (com.snc.ng_bsm) plugin, such as tables.

Note: The Application Files table lists the components that are installed with this application. For instructions on how to access this table, see [Find components installed with an application](#).

Tables installed

Table	Description
Available CI icons [ngbsm_ci_icons]	Stores all available CI class icons.
Icons for CI types [ngbsm_ci_type_icon]	Maps icons to CI class names.
Map Script [ngbsm_script]	Custom scripts that run in real time and generate a custom view of a map for a specific CI.
Map View [ngbsm_view]	Serialized map views saved by users.
Map Filter [ngbsm_filter]	Filters saved by users.
Menu Action [ngbsm_context_menu]	Default and custom context menu actions that appear when users right click a map.
Related Item [ngbsm_related_item]	Stores which reference fields should be treated as relationships when building the map. This allows users to include CI's that are related via a reference field instead of a relationship.

Table	Description
Edge Colors [bsm_edge_color]	Color definitions to use when drawing the relationships between nodes based on relationship type.
Map Indicator [bsm_indicator]	Stores all map indicators.
BSM Saved Map [bsm_graph]	Details of maps.
BSM Map Actions [bsm_action]	Actions on the map.
BSM Map View [map_view]	Parents' predefined filters.
Map View Configuration Types [map_view_ci_type]	Configuration type filters, limiting the CI class types to be displayed, per predefined filter.
Map View Relationship Types [map_view_rel_type]	Relationship type filters, limiting the links to be displayed between CIs, per each predefined filter.
[map_viewroles]	Roles that a specific predefined filter should be applied to.
CI Filters	CI attribute filters, limiting the CIs to be displayed, per predefined filter.

Table	Description
[map_filter]	

Querying the CMDB

The CMDB Query Builder allows you to easily build complex infrastructure and service queries, that span multiple CMDB classes, non-CMDB tables, and that involve many CIs that are connected by different relationships.

The CMDB Query Builder provides a canvas into which you drag the CI classes that you want to include in a query. Then you add relationships, AND/OR operators between the CI classes, and define the relationship properties to query for. You can use saved queries to populate a CMDB group with CIs, and then use scriptable APIs to retrieve the CI list and apply actions collectively to all the CIs in the group.

There are two query types: CMDB Query and a Service Mapping query, which you can use separately or in combination to create queries such as:

- All hardware in my service offering that has Windows installed.
- All CIs of a certain type in an application service. For example, all Apaches/Web Servers/Linux servers per service.
- All virtual servers and the physical servers that host them.
- All servers that are not mapped to any application service.
- All application services and their associated servers and the cost of each server. This query helps evaluate the cost of technology for each application service.

Starting node: The starting point of the query which is labeled as **STARTING NODE** on the Query Builder canvas. The first class that you drag to the canvas becomes automatically the starting node of the query and you cannot select a different starting node. In a complex query, the starting node must always be the only node connected to an AND/OR operator. If you try to connect a second node to an operator that the starting node is connected to, the query fails to run and a prompt to select a different starting node appears.

Additional information

- For a webinar, see [CMDB Query Builder Queries and Reporting - Platform Analytics Academy](#) blog post in the ServiceNow Community.
- For some tips and basic troubleshooting for the CMDB Query Builder, see the [CMDB Query Builder \[KB0681251\]](#).

Intelligent Search for CMDB integration

By default, [Intelligent Search for CMDB](#) functionality is integrated into the CMDB Query Builder. When opening the Query Builder, you can use the Intelligent Search search box which appears above the Query Builder canvas. Intelligent Search lets you use everyday natural language query (NLQ) to build a query. Intelligent Search parses, resolves any ambiguities in table names and relationship types, and then converts your search string into a valid query. The query appears fully constructed on the Query Builder canvas where you can run or continue and develop the query.

The integration of Intelligent Search for CMDB with the CMDB Query Builder is controlled by the system property `glide.cmdb.query.nlq.activated`, which is set to **true** by default. If you set the property to **false**, Intelligent Search for CMDB will not be available within the Query Builder.

CMDB Query

A query type that queries the infrastructure for CI classes and the relationships and references that connect them. You can optionally add the context of non-CMDB tables to a CMDB query.

You can include [Application services](#) in a CMDB query, to find, for example:

- All critical application services in your database.
- All infrastructure in a particular application service.
- All incidents for a particular CI in an application service, or all incidents for all the CIs of an application service.
- All application services with a pattern of a service connected to a database, and where the database has incidents.

The list of available non-CMDB tables includes a subset of tables within the system, which have a reference to the Configuration Item [cmdb_ci] class or its children. The list of non-CMDB tables, includes tables such as Asset, Task, and Problem. You can use the system property `glide.cmdb.query.non_cmdb.black_listed_tables` to narrow down the list of non-CMDB tables to choose from.

Service Mapping Query

A query type that queries application services. The query is framed within an application service map. You define a pattern, and query for application service maps that have that pattern in their definition. The relationships in Service Mapping queries are matched by single-level direct relationships which is similar to the CMDB queries, and in addition, they are also matched by multi-level indirect relationships if they exist. A query for a relationship between two CI classes is satisfied even if the two CI classes are connected by intermediate CI classes that are not specified in the query.

Combination Query

You can combine the two query types by incorporating a saved Service Mapping query into a CMDB query. For example, create a CMDB query for Windows Servers that are connected to Tomcat WAR. Then connect the Tomcat WAR CI class to a Service Mapping query. The query changes to find Windows Servers that are connected to Tomcat WAR which is included in the services that returned by the Service Mapping query. You can inverse that query by choosing **Does Not Belong To Service**. This changes the query to find Windows Servers that are connected to Tomcat WAR that is not included in services returned by the Service Mapping query.

Relationship properties

When you connect CI classes on the canvas, the CMDB Query Builder displays the Connection Properties in the right-side bar, where you can configure the properties of the relationship, such as the relationship direction. For Service Mapping queries, you can configure whether to query for related or unrelated CIs.

Connection properties include:

- Relationship type: Query for CIs and descending classes with specific relationship types.

- Relationship direction: Which CI class is the parent and which CI class is the child in the relationship.
- Relationship level: Query only on first-level relationship or also on second-level relationships.
- No relationships: Query for CIs which have no relationship to the set class.
- References fields: A field that the parent and ancestor parent CI classes use to reference the child CI class.

Newly added relationships between CI classes may take up to 30 minutes to appear in the relationship list.

- [Domain separation and CMDB Query Builder](#)

Domain separation is supported in the CMDB Query Builder. Domain separation enables you to separate data, processes, and administrative tasks into logical groupings called domains. You can control several aspects of this separation, including which users can see and access data.

- [Build a CMDB query using the CMDB Query Builder](#)

A CMDB query type that queries the infrastructure for CI classes and optionally non-CMDB tables, and the relationships and references that connect them.

- [Build a Service Mapping query using the CMDB Query Builder](#)

The Service Mapping query type is a pattern consisting of classes and relationships between those classes. After you build the pattern and run the query, the query returns all the Service Mapping services that contain that pattern.

- [Sample queries](#)

Use the following sample queries to build your own CMDB queries and Service Mapping queries.

- [Run a partial CMDB query](#)

You can run a partial query in the CMDB Query Builder by defining a section of a query, and then running it.

- [Delete a CMDB query](#)

Delete a CMDB query that is no longer used or needed.

- [Batch size for CMDB Query Builder queries](#)

In a base system, a global batch size of 100 is allocated for every Query Builder query run. If needed, you can use a system property to override the default global batch size, or optimize the batch size value per saved query.

- [Navigation in CMDB Query Builder](#)

Use the navigation tools to enlarge or shrink the query, to move the query, or to border a section of the query to run.

- [Create reports in CMDB Query Builder](#)

Use CMDB Query Builder reports to show the results of a CMDB query or a Service Mapping query. Create a basic report, or a dynamic report that automatically updates when the results of the associated saved query change.

- [Search saved queries](#)

The CMDB Query Builder allows you to search for a specific saved query using any combination of search criteria such as the query's name, type, custom tags, and who created or updated the query.

- [Create a schedule for a CMDB query](#)

Schedule a saved CMDB query to run once at a scheduled time or on a recurring schedule, and to email the query results to specified users.

- [Export and import a CMDB query](#)

Export a saved CMDB or Service Mapping query definition to an XML file which you can later import and run in the CMDB Query Builder. This process lets you port a saved query between instances, such as from a development environment to a production environment.

- [Settings for CMDB Query Builder](#)

Use settings to control some aspects of the CMDB Query Builder behavior.

- [Properties for CMDB Query Builder](#)

Use the CMDB Query Builder properties to configure query processing.

Related concepts

- [CMDB groups](#)

Domain separation and CMDB Query Builder

Domain separation is supported in the CMDB Query Builder. Domain separation enables you to separate data, processes, and administrative tasks into logical groupings called domains. You can control several aspects of this separation, including which users can see and access data.

Overview

How domain separation works in the CMDB Query Builder

With the CMDB Query Builder you can easily build complex infrastructure and service queries that span multiple CMDB classes, and that involve many CIs that are connected by different relationships. Domain separation is set to be on by default.

- **Saved Query**

The user creates a query by dragging a class node from the class hierarchy and dropping it to the canvas and connecting the nodes with the relationships type.

The user can save the created query as an XML file to the database [qb_saved_query] table in the CMDB for future use. The saved query is domain separated.

- **Query results**

With a saved query, the user clicks **Run** and the query result is saved and displays in the platform list view.

In the query results, the domain separation behaves in the same way as the platform list view for the CI relationship [cmdb_rel_ci] table and CMDB CI [cmdb] table. Consequently, since the CI relationship

is not domain separated, all relationships of the query result display, regardless of the domains. Conversely, if the query result is CI only, since the CMDB CI is domain separated, the results display only if visible in the current domain.

Related concepts

- [Domain separation and Configuration Management Database \(CMDB\)](#)

Build a CMDB query using the CMDB Query Builder

A CMDB query type that queries the infrastructure for CI classes and optionally non-CMDB tables, and the relationships and references that connect them.

Before you begin

The [Core UI plugin](#) (com.glide.ui.ui16) must be activated.

Role required: cmdb_query_builder_read to only view and run saved queries, and cmdb_query_builder (contained for itil, itil_admin, and asset) to create and save queries, modify saved queries, and run queries.

Authorized users can update and [delete](#) a query created by another user.

About this task

Build the query by dragging the CI classes and non-CMDB tables that you want to include in the query. Then dropping them as nodes on the canvas, and defining relationship properties between them. You can filter on the attributes of any node to narrow down the results to a specific set of CIs of that class or to a single specific CI. You can also select which property columns appear in the query results.

As you step through building a query, list options and other user interface elements of the CMDB Query Builder, are dynamically filtered as appropriate to your selections.

To learn more about using Query Builder, see the [CMDB Query Builder Queries and Reporting - Platform Analytics Academy ServiceNow Community](#) video. For a step-by-step walk through of building CMDB

queries in the Query Builder, including queries with application services, see [Sample queries](#).

Procedure

1. Navigate to **All > Configuration** and click **CMDB Query Builder**.
 2. On the **CMDB Query Builder** page do either of the following steps:
 - Click **Create new**. Type in a **Name**, choose **CMDB Query** as the **Query type**, and then click **Create**.
 - Click a widget of a saved query to continue building an existing query. [Search saved queries](#) first if needed.
 - Point to the upper right corner of a saved query widget, and click the **Duplicate Query** icon to edit a copy of a saved query. The default name of the new query contains the string 'copy'.
 3. On the canvas, you can do any of the following operations:
 - Add CI classes to the query: On the **CMDB Classes** tab, select classes from the hierarchy list and drag them to the canvas.
 - Add an Application Service CI to the query: On the **CMDB Classes** tab, select the **Application Service** class from the hierarchy list and drag it to the canvas.

In Application Service Properties on the right-side bar, you can select **Convert attached nodes to pattern** to query on patterns between the application service and other CMDB class nodes. When querying on a pattern, the nodes on both ends of the pattern connection, can be any number of levels apart. If **Convert attached nodes to pattern** is not selected, then the connection between the application service node and other CMDB class nodes, represent direct relationships.
 - Add non-CMDB tables to the query: Select a table from the **Non-CMDB Tables** list and drag it to the canvas.
- Note:** A non-CMDB table cannot be the starting node in the query.

- Add connections (relationships or patterns for application services) between two nodes on the canvas:
 - a. On the first node in the connection, click the small square at the center of the right side.
 - b. On the second node in the connection, click the small square at the center of the left side to create the connection.

Connection UI Notations

Notation	Description
Full line	A relationship in a CMDB query.
Red asterisk at the center of the connection line	Information such as relationship type is missing, invalidating the query.
Levels:<n> Types:<n> or a <Reference type> notation on the connection line	As applicable: The number of relationship levels and the number of relationship types included for the connection. Or, a reference type for a relationship that is a reference.
Dashed line	A pattern connection between an application service node and another node.

- In Connection Properties on the right-side bar, configure relationship settings (click the connection line if necessary):
 - a. In the Relationship Direction section, select the **Parent** node (the **Child** node automatically adjusts).
 - b. In the Relationship Levels section, set **Level to First level relationships** if the Cls are directly connected. Or, **Up to 2nd**

level relationships if the CIs are connected either directly or indirectly through another CI.

C:

In the Relationship Types and Related Items section, select either option:

Option	Description
No Relationships	To query for CIs with no connecting relationships, such as All Tomcat WAR CIs which are not connected to a Windows Server.
Add Relationship Types	To select specific or any relationship type.
Add a Related Item	To query for related CIs between the nodes.

- Configure CI reference column for a connection to a non-CMDB table: In the Connection Properties right-side bar, in the CI Reference Column section, select the column with a reference to a CI from the **Use CI reference column** list. If only one option is available, it is automatically selected.
- Configure the pattern between an application service node set with the **Convert attached nodes to pattern** option, and a non-CMDB table node: Select **Apply <table> reference filter to all nodes in the pattern** to apply the query to the application service CI itself and to the CIs within the application service.
- Add filters to a class node: Apply filters to narrow down a class query to a specific set of CIs or to a single specific CI.
 - Point to the node to add a filter to, and then click the **Apply filters** icon that pops up above the node.
 - In the Filters section, add attribute and **related list conditions**.
 - Close the **Filters** section.

For example: Add a filter for database location to query for databases located in Seattle.

Click **Applied Filters** in the right-side bar to view all filters for each node on the canvas.

- Add And/Or operators to the query:

a: Connect one node to two other nodes.

b: Click the **And** box that appears on the connection line, to toggle between the **And/Or** operators.

For example: C1 is Tomcat WAR, C2 is Linux Server, and C3 is Windows Server. Query for All Tomcat WAR CIs which are connected either to Linux Server Or to a Windows Server.

- Add property columns for a node, to appear in the query results:

Note: For a relationship, the query results include the parent, child, and type columns. You cannot add any other columns from the [cmdb_rel_ci] table.

a: Click **Properties** in the right-side pane.

b: Click a node once or twice, so that the Report Columns section appears in the right-side bar, and then click **Add Columns**.

c: Select properties and then click outside the properties list to close it.

- Create a combination query by integrating a Service Mapping query into a CMDB query:

a: When building a CMDB query, click **Saved Service Queries** in the left-side bar.

b: Select and then drag a Service Mapping query to the canvas.

This query returns all CIs that satisfy the CMDB query, and that are included in the services returned by the embedded Service Mapping query.

- Add a search tag that can then be used as a search criteria for saved queries:
 - a: Click the **Add Tags** icon at the top of the canvas.
 - b: Click **Add Tag** and in the **Query Tags** dialog box enter one or more tag strings.
 - c: Click the **Add Tags** icon again to close the **Query Tags** dialog box.
- 4. Click **Save**.

On the **Saved Queries** tab, point to a saved query widget and click **Query Information**. Query details such as the query type, last update date, CMDB groups associated with the query, and the query schedules appear.

What to do next

-

Click **Run**.

Only the first 100 results of the query appear in the results pane.

- Click **Load More Results** to view the next set of 100 results.
- Click **Load All Results** to view the rest of the query results, up to the number specified by the `glide.cmdb.query.max_results_limit` system property (10,000 by default).

Click a CI to open its CI form, and on the CI form click **Dashboard** to view CI health in the CI dashboard.

Note:

- Ensure that the `glide.security.use_csrf_token` property is set to true, allowing all results to appear.
- When **Level** is set to **Up to 2nd level relationships**, the relationship type does not appear in the query results.
- When a query is running, wait for it to complete or to time out before opening or running another query.

- **Modify Query Builder settings:** Click the  (**Settings**) icon to open the **Query Builder Settings** dialog box.
- Copy and share the URL of a saved query with users that have access to the CMDB Query Builder. Pasting the shared URL in a new internet browser window, directly opens the saved query in the CMDB Query Builder.
- Create reports in CMDB Query Builder.
- Create a schedule to run the query at a future time, and to email the results to interested parties.
- Export query results:

Click the Query Results context menu and select **Export**. Even if the **Load More Results** button is visible, indicating that there are additional query results, only the results that are visible are exported.

- Export and import a CMDB query to port a query definition between instances.
- Populate a CMDB group using a saved query.
- Delete a CMDB query.

Related concepts

- Sample queries

Build a Service Mapping query using the CMDB Query Builder

The Service Mapping query type is a pattern consisting of classes and relationships between those classes. After you build the pattern and run the query, the query returns all the Service Mapping services that contain that pattern.

Before you begin

The [Core UI plugin](#) (com.glide.ui.ui16) must be activated.

Role required: cmdb_query_builder_read to only view and run saved queries, and cmdb_query_builder (contained for itil, itil_admin, and asset) to create and save queries, modify saved queries, and run queries.

Authorized users can update and [delete](#) a query created by another user.

About this task

Build the query by dragging the CI classes that you want to include in the query, dropping them as nodes on the canvas, and then defining relationship properties between them. For every class node in the query, you can filter on its attributes to narrow down the results to a specific set of CIs of that class or to a single specific CI. You can also select which property columns appear in the query results.

As you step through building a query, list options and other user interface elements of the CMDB Query Builder, are dynamically filtered as appropriate to your selections.

See [Sample queries](#) for a step-by-step walk through of building a Service Mapping query in the CMDB Query Builder.

Procedure

1. Navigate to **All > Configuration > CMDB Query Builder**.
2. On the **CMDB Query Builder** page do either of the following:
 - a. Click **Create new**. Type in a **Name**, choose **Service Mapping Query** as the **Query type**, and then click **Create**.

- b. Click on a widget of a saved query to continue building an existing query. [Search saved queries](#) first if needed.
 - c. Point to the upper right corner of a saved query widget, and click the **Duplicate Query** icon to edit a copy of a saved query. The new query's default name contains the string 'copy'.
3. On the canvas, you can do any of the following:
- Add CI classes to the query: Select classes from the **CMDB Classes** hierarchy list and drag them to the canvas.
 - Add connections (relationships) between two nodes on the canvas:
 - a: On the first node in the relationship, click the small square at the center of the right side.
 - b: On the second node in the relationship, click the small square at the center of the left side to create the connection.
 - c: In **Connection Properties** on the right-side bar, configure the following (click the connection line if necessary):
 - In the Relationship Direction section, select the **Parent** node (the **Child** node automatically adjusts).
 - In the Service Query Properties section, select **Find Related CIs** or **Find Unrelated CIs** to query for a pattern in which the two classes have or do not have relationships with each other, respectively.

For example, All Tomcat WAR CIs which are not connected to a Windows Server.

Relationship UI Notations

Notation	Description
Dashed line	A relationship in a Service Mapping query.

- Add filters to a class node: Apply filters to narrow down a class to a specific set of CIs or to a single specific CI.

- a. Point to the node to add a filter to, and then click the **Apply filters** icon that pops up above the node.

- b. In the Filters section, add attribute and [related list conditions](#).

- c. Close the **Filters** section.

For example: Add a filter for business criticality to query for businesses that are 'most critical'.

Click **Applied Filters** in the right-side bar to view all filters for each node on the canvas.

- Add And/Or operators to the query:

- a. Connect one node to two other nodes.

- b. Click the **And** box that appears on the connection line, to toggle between the **And** and the **Or** operators.

For example C1 is Tomcat WAR, C2 is Linux Server, and C3 is Windows Server. Query for all Tomcat WAR CIs which are connected either to Linux Server Or to a Windows Server.

- Add property columns for a node, to display in the query results:

Note: For a relationship, the query results display the parent, child, and type columns. You cannot add any other columns from the [cmdb_rel_ci] table.

- a. Click **Properties** in the right-side pane.

- b. Click a node once or twice, so that the node's Report Columns section appears in the right-side bar, and then click **Add Columns**.

- c. Select properties and then click outside the properties list to close it.

- Select columns and add filters that will be applied to the resulting set of services:

- a. Select **Properties** in the right-side bar, and then click an empty space on the canvas to ensure that nothing is selected.

- b. Click **Add Columns** at the bottom of the right-side bar and select columns to add. Click outside the columns list to close it.
 - c. Click the **Apply Service Mapping Query Filters** icon at the top of the canvas and add filters.
 - Inverse the entire query and search for all Service Mapping services that do not include the query pattern: Click a node once or twice so that Query Properties appear in the right-side bar. In the Metadata section, toggle **Services Including This Pattern** to enable or disable the option.
 - Add a search tag that can then be used as a search criteria for saved queries:
 - a. Click the **Add Tags** icon at the top of the canvas.
 - b. Click **Add Tag** and in the **Query Tags** dialog box enter one or more tag strings.
 - c. Click the **Add Tags** icon again to close the **Query Tags** dialog box.
4. Click **Save**.

On the **Saved Queries** tab, point to a saved query widget and click **Query Information** to view query details such as the query type, last update date, and the query schedules.

What to do next

-

Click **Run**.

The query results pane displays only the first 100 results of the query.

- Click **Load More Results** to display the next set of 100 results.
- Click **Load All Results** to display the rest of the query results, up to the number specified by the `glide.cmdb.query.max_results_limit` system property (10,000 by default).

Click a CI to open its CI form, and on the CI form click **Dashboard** to view CI health in the CI dashboard.

Note: When a query is running, wait for it to complete or to time out before opening or running another query.

- [Modify Query Builder settings](#): Click the  (**Settings**) icon to open the **Query Builder Settings** dialog box.
- Copy and share the URL of a saved query with users that have access to the CMDB Query Builder. Pasting the shared URL in a new internet browser window, directly opens the saved query in the CMDB Query Builder.
- Create reports in [CMDB Query Builder](#).
- Create a [schedule](#) to run the query at a future time, and to email the results to interested parties.
- Export query results:

Export query results that are visible: Click the Query Results context menu and select **Export**. Even if the **Load More Results** button is visible, indicating that there are additional query results, only the results that are visible are exported.

- [Export and import a CMDB query](#) to port a query definition between instances.
- [Populate a CMDB group](#) using the saved query.
- [Delete a CMDB query](#).

Related concepts

- [Sample queries](#)

Sample queries

Use the following sample queries to build your own CMDB queries and Service Mapping queries.

Using the CMDB Query Builder requires that the [Core UI plugin](#) (com.glide.ui.ui16) is activated.

CMDB query sample

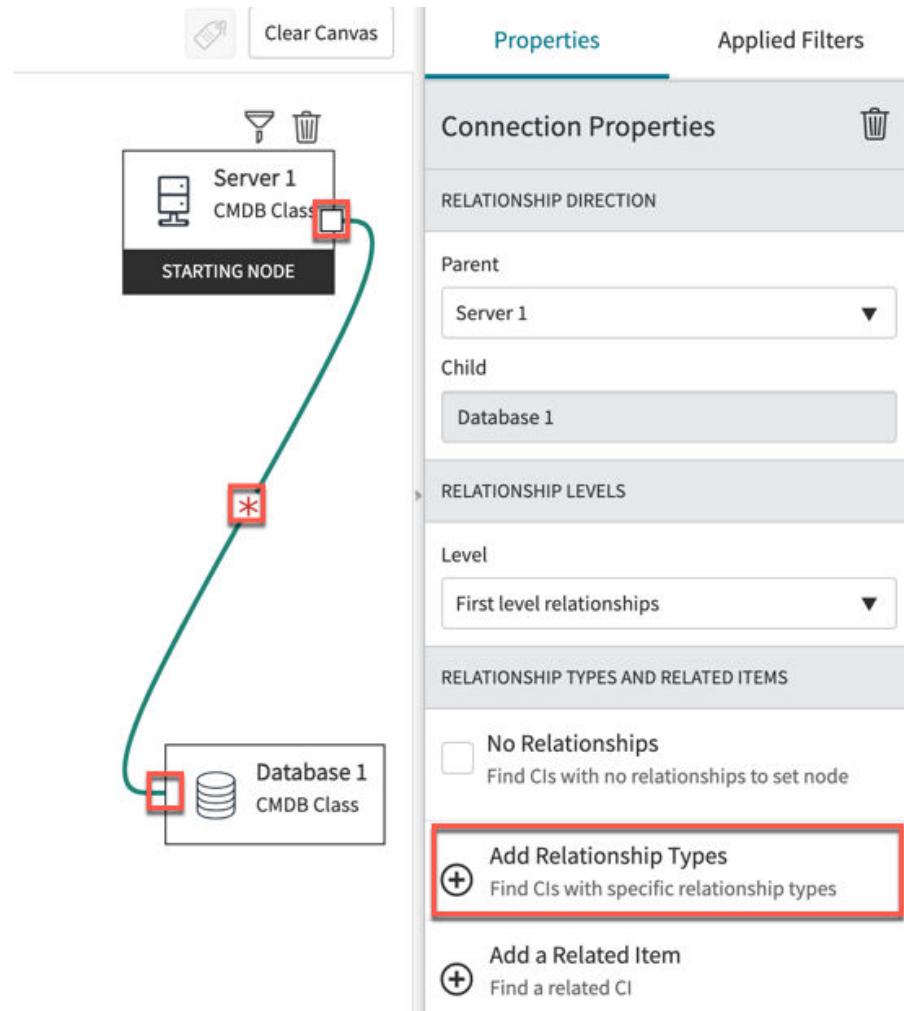
Use this example to build a CMDB query to find all servers with a connection to a database.

Before you begin

Role required: none

Example

1. Navigate to **All > Configuration > CMDB Query Builder**.
2. Click **Create new**. Enter a **Name** - All servers with a connection to a DB. Choose **CMDB Query**, and click **Create**.
3. In the **CMDB Classes** list, locate the **Server** class, and drag it to the canvas.
4. Locate the **Database** class, and place it to the right of the **Server** class node on the canvas.
5. Click at the center of the right side of **Server**, and then at the center of the left side of **Database** to create a connection line between the two class nodes.
6. Click once or twice on the connection line until the Connection Properties panel appears in the right-side bar. In the Relationship Types and Related Items section, click **Add Relationship Types** and add all the relationships from the list.



Settings in the Relationship Direction section reflect the parent-child direction in the relationship. If the Database class is the parent in the relationship, then the **Parent** and **Child** settings are switched.

7. Click **Save**, and then click **Saved Queries** on the left to see the widget for the saved query.
8. Click the query widget to return to the canvas in edit mode.
9. Click **Run** to execute the query.

Review the query results. Each row displays the name of a server CI, the name of a database CI, and the relationship type between them.

10. Add columns to the query results:

- a. Click the **Server 1** node on the canvas once or twice so that the Server 1 Report Columns section appears in the right-side pane. Click **Add Columns**.
- b. Select **Manufacturer** and then click outside the columns list to close it.
- c. Click **Run**.

Review the query results which now include the **Manufacturer** column.

- d. Click **Save** again to save all your customization for this query.

CMDB query sample - Application service 1

Use this example to build a CMDB query to find all critical application services, and their owner.

Before you begin

Role required: none

Example

1. Navigate to **All > Configuration > CMDB Query Builder**.
2. Click **Create new**.
3. Enter **All critical application services** as the query **Name**. Choose **CMDB Query** and then click **Create**.
4. In the **CMDB Classes** list, locate the **Application Service** class, and then drag it to the canvas.
5. Add a filter to the application service node:
 - a. Point to the application service node, and then click the **Apply filters** icon that appear.

- b. In the Filters section, add the condition **[business criticality] [is] [1 - most critical]**.
 - c. Close the Filters section.
6. Add columns to the query results:
 - a. In the Properties right-side bar, click **Add Columns**.
 - b. Select **business criticality** and **owned by**, and then click outside the columns list to close it.
 7. Click **Save**.
 8. Click **Run** and then review the results. You can for example, locate any of the critical application services without an owner.

CMDB query sample - Application service 2

Use this example to build a CMDB query to find all application services, for which there is an incident or a change request, for either, the application service itself, or any CI within the service.

Before you begin

Role required: none

Example

1. Navigate to **All > Configuration > CMDB Query Builder**.
2. Click **Create new**.
3. Enter **Application services with incidents or change requests** as the query **Name**. Choose **CMDB Query** and then click **Create**.
4. In the **CMDB Classes** list, locate the **Application Service** class and then drag it to the canvas.
5. Click **Non-CMDB Tables**.
6. Locate the **Incidents** class in the class hierarchy, and then drag it to the canvas.

7. Locate the **Change Requests** class in the class hierarchy, and then drag it to the canvas.
8. Connect the Application Service and the Incidents nodes, and then, in the Properties right-side bar:
 - a. Select **Apply Incidents reference filter to all nodes in the pattern**.
 - b. Set **Use CI reference column to Configuration item**.
9. Connect the Application Service and the Change Request node, and then, in the Properties right-side bar:
 - a. Select **Apply Change Request reference filter to all nodes in the pattern**.
 - b. Set **Use CI reference column to Configuration item**.
10. Click the **And** operator between the Incidents and the Change Request nodes, and switch it to **Or**.
11. Click **Save**.
12. Click **Run** and then review the results.

CMDB query sample - Application service 3

Use this example to build a CMDB query to find all hardware in my service offering that has Windows installed.

Before you begin

Role required: none

Example

1. Navigate to **All > Configuration > CMDB Query Builder**.
2. Click **Create new**.
3. Enter All hardware in my service offering that has Windows installed as the query **Name**. Choose **CMDB Query** and then click **Create**.

4. In the **CMDB Classes** list, locate the following classes, and then drag them to the canvas.

- **Service**
- **Service Offering**
- **Application Service**
- Searching for infrastructure, **Hardware**

5. Connect the Service node to the Service Offering node.

In the Properties right-side bar, click **Add Relationship Type** and select the **Connect to::Connected by** relationship.

6. Connect the Serviced Offering node to the Application Service node.

In the Properties right-side bar, click **Add Relationship Type** and select the **Connect to::Connected by** relationship.

7. Click the Application Service node.

In the Properties right-side bar, select **Convert attached nodes to pattern** to include all CIs within the application service, in the query.

8. Connect the Application Service node to the Hardware node.

9. All infrastructure under Service,

10. Click **Save**.

11. Click **Run** and then review the results.

You can click **Column options** of the Service column header, and select to **Group by Service**. Then expand a service to see all the hardware infrastructure under that service.

12. Return to the CMDB Query Builder window, to expand the query to include only infrastructure CIs on which Windows is installed.

13. Click **Non-CMDB Tables**, locate the **Software Instance** class, and drag it to the canvas.

14. Connect the Hardware node to the Software Instance node.

In the Properties right-side bar, set **Use CI reference column** to **Installed on**.

15. Point to the Software Instance node, and click on the **Apply filters** icon that appears. In the Filters section, add the condition **[Product Name.Name] [is] [windows]**. Close the Filters section.
16. Click **Save**.
17. Click **Run** and review the new results.

Service Mapping query sample

Use this example to build a Service Mapping query to find all Linux servers in services.

Before you begin

Role required: none

Example

1. Navigate to **All > Configuration** and click **CMDB Query Builder**
2. Click **Create new**. Enter a **Name - Linux server in services**. Choose **Service Mapping Query**, and click **Create**.
3. In the **CMDB Classes** hierarchy list, locate **Linux Server** and drag it to the canvas.
4. Click **Run**.

Review the query results. Each row displays the name of a Service Mapping Service and the name of a Linux Server that is a member of that service.

5. On the right-side pane, click **Disable Service Including This Pattern** and then click **Run** again.

Review the query results. Now, each row displays the name of a Service Mapping Service that does not include the specified Linux Server.

Run a partial CMDB query

You can run a partial query in the CMDB Query Builder by defining a section of a query, and then running it.

Before you begin

The [Core UI plugin](#) (com.glide.ui.ui16) must be activated.

Role required: cmdb_query_builder (contained for itil, itil_admin, and asset)

About this task

While building a query or reviewing a saved query, you can run only a section of the query. On the canvas in the CMDB Query Builder, highlight a section of the query which contains the nodes and relationships of the partial query that you want to run. You can then examine the results of the partial query, and update the query if needed.

Procedure

1. Navigate to **All > Configuration > CMDB Query Builder**.
2. On the CMDB Query Builder page, click a tile to open an existing query.
3. Click the selection tool under the [navigation tool](#) to switch to a section selection mode.
4. Border a section of the query:
 - a. Click the mouse device on the upper left corner of the section that you want to create.
 - b. Drag the mouse device to the bottom right corner of the section that you want to create. As you drag the mouse device, the selected section is highlighted in light blue.
 - c. Release the mouse device. The query nodes that are included in the partial query, appear with a blue border.
5. Click **Run**.

6. In the Pick Starting Node dialog box, select the starting node for the partial query, and click **Confirm**.

Result

The results of the partial query appear in the Results pane.

Delete a CMDB query

Delete a CMDB query that is no longer used or needed.

Before you begin

The query that you want to delete must be already saved.

Role required: cmdb_query_builder (contained for itil, itil_admin, and asset). Authorized users can delete a query created by another user.

Procedure

1. Navigate to **All > Configuration** and click **CMDB Query Builder**.
2. On the **CMDB Query Builder** page, select the Saved Queries tab and set the viewing mode to Card view ().
3. Hover over the card with the query that you want to delete.
4. Click the Delete Query (X) icon that appears in the upper right corner of the card.

Batch size for CMDB Query Builder queries

In a base system, a global batch size of 100 is allocated for every Query Builder query run. If needed, you can use a system property to override the default global batch size, or optimize the batch size value per saved query.

Queries can differ widely as they can be configured to query a wide variety of classes. Therefore, the batch size in the base system might not be optimal for every query, and some queries might time out or take a long time to complete. The optimal batch size for running queries

depends on system load such as amount of data and number of relationships in your system. Contact Support for assistance in calculating the batch size for your query.

Batch size is applicable and behaves the same in all query run scenarios, regardless of how the run was initiated:

- Query Builder user interface (ad hoc or saved query)
- [Query Builder Scriptable API](#)
- Scheduled jobs
- [CMDB groups](#)

The batch size for query runs is allocated in the following priority order:

1. The value in the Execution Batch Size field in the Saved Queries table, for a specific saved query. If set, this value applies only to the saved query, and has priority over the global value of 100 and the value of the [glide.cmdb.query.batch_size](#) system property.
2. The value of the system property [glide.cmdb.query.batch_size](#), if exists, determines globally the batch size that is allocated to all query runs. If you add and set this property, the value applies to all queries, other than saved queries with Execution Batch Size value set.
3. A global value of 100, if the previous two options are not configured.

Modifying batch size for queries

If you are experiencing performance problems when running queries, you can modify the batch size value:

- Globally for all queries: By adding (if necessary) and setting the value of the system property [glide.cmdb.query.batch_size](#).
- Per saved query: [Set batch size for a specific saved query](#)

Set batch size for a specific saved query

Configure a custom batch size for a CMDB Query Builder saved query that takes a long time to complete or that times out. A custom batch size overrides the global batch size in the base system and the value of the [glide.cmdb.query.batch.size](#) system property.

Before you begin

Role required: admin

About this task

Contact Support for assistance in calculating the batch size for your query.

Procedure

1. In the Filter navigator, enter `qb_saved_query.list` and press Enter to navigate to the Saved Queries table.
2. In the Saved Queries list view, locate the saved query for which you want to change batch size.
3. Set or modify the value in the Execution Batch Size field.
Set the value to be greater than the global value in the `glide.cmdb.query.batch.size` property, or increase any existing value.

Navigation in CMDB Query Builder

Use the navigation tools to enlarge or shrink the query, to move the query, or to border a section of the query to run.

Using the Query Builder requires that the [Core UI plugin](#) (`com.glide.ui.ui16`) is activated.



Use the buttons in the navigation tool as follows:

- Use the plus sign (+) to increase magnification of the query.
- Use the minus sign (-) to decrease magnification of the query.
- Click the center dot to center the query on the canvas.

- Use the direction arrows to move the query in that direction.
- Use the selection tool under the navigation tool to toggle between two states:
 - Moving the entire query on the canvas.
 - Bordering a section of the query, which you can then run as a partial query.

Create reports in CMDB Query Builder

Use CMDB Query Builder reports to show the results of a CMDB query or a Service Mapping query. Create a basic report, or a dynamic report that automatically updates when the results of the associated saved query change.

Create a dynamic report

After running a saved query in the CMDB Query Builder, create a dynamic report that continuously updates to show the latest query results. You can use a dynamic report as any other report created using Reporting and you can add it to Performance Analytics dashboards.

Before you begin

Ensure that the query that you want to create a dynamic report for is a saved query, and that there is a specified schedule for the query. Also, run the saved query and ensure that all query results are visible.

Role required:

- To create: cmdb_query_builder and report_user
- To view: Reporting role requirements might apply, see [Administering reports](#) for Reporting role requirements.

In a base system, the cmdb_query_builder role is contained in the itil and asset roles.

About this task

The **Create Report** button in CMDB Query Builder which is used to create a dynamic report, is activated only if:

- The query is saved
- The query has a schedule
- The entire set of query results is present after a query run

The initial dynamic report that you create, is based on the results from the initial run of the saved query. Then, on every subsequent run of the saved query, the associated report automatically updates with the latest query results.

However, if you change the query definition itself, the query and the report are no longer in sync and you must create a new report.

Procedure

1. Navigate to **All > Configuration > CMDB Query Builder**.
2. In the **Saved Queries** tab, select the saved query for which you want to create a report.
Ensure that the query has a specified schedule.
3. Click **Run** and ensure that all query results appear. Click **Load All Results** if available, to load all results.
The **Create Report** button is enabled only if all query results are showing.
4. Click **Create Report**.
If the CMDB Query Builder displays the query results in a new tab, then after the new tab opens with the query results, return to the CMDB Query Builder window.
5. In the Report Designer, click **Next** or **Back** to view and configure the new report in the **Data**, **Type**, **Configure**, and **Style** tabs.
The report is pre-populated with the CMDB query results and a few other report details.
 - **Report name** is set to the name of the saved query.
 - **Source type** is set to **Data source**.
 - **Data source** is set to the table in which the query results are stored.
 - **Query Sys ID** is the ID of the latest run of the query.

For more details about Reporting and about configuring a report in the Report Designer, see [Reporting, Creating reports](#).

6. Click **Save** or **Run**.

Result

CMDB Query Builder creates a report source which you can attach to a report and use with dashboards. For more details about report sources, see [Report sources](#).

What to do next

Use either of the following steps to view the new report source. The name of the new report source is set to the name of the CMDB query it was created from, and cannot be changed.

- In Query Builder, click **Saved Queries**. In the Saved Queries window, click the **Query Information** icon in the tile of the saved query. Scroll to the bottom of the information list and then click the link under **Report source**.
- Navigate to **All > Reports > Administration > Report Sources** and locate the new report source.

Create a basic report

After running a query in the CMDB Query Builder, you can create a basic report that is scoped to the query execution.

Before you begin

The [Core UI plugin](#) (com.glide.ui.ui16) must be activated.

Role required:

- View report: cmdb_query_builder or cmdb_query_builder_read
- Create report: cmdb_query_builder or cmdb_query_builder_read, and report_user

In a base system, the cmdb_query_builder role is contained in the itil and asset roles.

Procedure

1. Navigate to **All > Configuration** and click **CMDB Query Builder**.
2. Build a query.
3. In the query results pane, click **Load More Results** or **Load All Results** to load all the results that you want to include in the basic report.
4. In the query results pane, click the column context menu and select **Bar Chart** or **Pie Chart**.

Result

The Reports application creates a basic report, which is scoped to the query results that are currently loaded and is static.

Related tasks

- [Build a CMDB query using the CMDB Query Builder](#)
- [Build a Service Mapping query using the CMDB Query Builder](#)
- [Run a partial CMDB query](#)

Search saved queries

The CMDB Query Builder allows you to search for a specific saved query using any combination of search criteria such as the query's name, type, custom tags, and who created or updated the query.

Before you begin

To locate a saved query using a **Query Tags** search criteria, the query must have a query tag associated with it. For more information see [Build a CMDB query using the CMDB Query Builder](#), or [Build a Service Mapping query using the CMDB Query Builder](#).

The [Core UI plugin](#) (`com.glide.ui.ui16`) must be activated.

Role required: `cmdb_query_builder` (contained for `itil`, `itil_admin`, and `asset`)

Procedure

1. Navigate to **All > Configuration > CMDB Query Builder**.
2. On the Saved Queries tab, select a **Sort by** criteria.
3. In the **Search Saved Queries** field, enter a search string that corresponds to the **Sort by** selection and then select an item from the list that gets populated as you type.
The drop-down list includes search strings that can be applied to the sorting criteria such as 'Query name'.
4. Refine the search by entering additional **Search Saved Queries** search strings as needed.

Create a schedule for a CMDB query

Schedule a saved CMDB query to run once at a scheduled time or on a recurring schedule, and to email the query results to specified users.

Before you begin

The [Core UI plugin](#) (com.glide.ui.ui16) must be activated and a saved CMDB query that was built in the CMDB Query Builder must exist.

Role required: cmdb_query_builder (contained for itil and asset)

About this task

The query results are attached to the email as a file in the specified format. By default, the maximum result rows that can be attached is 10,000. This is controlled by a system property.

Procedure

1. If need to, navigate to **All > Configuration > CMDB Query Builder** and then click a saved query.
2. Click **Create Schedule** and fill out the form.

Scheduled Email of Query Builder form

Field	Description
Query	The query to run.
Users	Users who should receive query results email. To receive emails, users must have an Email address defined and have Notifications set to Enable in their user records.
Groups	Groups to email the query results to.
Zip output	Indicates whether the report should be sent as a zip file.
Active	Indicates whether to run the query according to the specified schedule.
Run	Frequency for running the query.
Time	Time of day to run the query.
Conditional	Indicates whether to display the Condition field, which allows you to specify conditions under which the query runs.
Omit if no records	Indicate whether to distribute email if the query returns zero results.
Email addresses	Email addresses of users who should receive the email but who are not in the system.

Field	Description
Subject	Text that appears in the subject line of the distribution email.
Introductory message	Additional message that is delivered with the query results.
Condition	User-created script that checks for certain conditions to be true before running the query. This field is visible only when Conditional is checked.
Type	File format to use for the query results. Note: Configure the form layout to add this field to the form.

Export and import a CMDB query

Export a saved CMDB or Service Mapping query definition to an XML file which you can later import and run in the CMDB Query Builder. This process lets you port a saved query between instances, such as from a development environment to a production environment.

Before you begin

- You must save a query before you can export it.
- Domain in an exported query must be visible in both, source and destination instances.

Role required: cmdb_query_builder (contained in itil, itil_admin, and asset).

About this task

When exporting a combination query, the integrated Service Mapping query definition is included in the exported query.

For backward compatibility, you can alternatively [Export and import a query as an update set](#).

Procedure

1. Navigate to **All > Configuration > CMDB Query Builder**.
2. Export a saved query:
 - a. In the **Saved Queries** tab, in either list view or card view, select a saved query.
 - b. Click the **Export query** icon at the top of the Query Builder canvas.
 - c. Wait for the **Query Export Complete** message to appear and then click **Download**.
You can now access the query XML file.
3. Import a saved query:
 - a. In the **Saved Queries** tab, click the **Import query** icon at the top of the CMDB Query Builder window.
 - b. In Finder, select the saved query XML file and click **Open**.
The imported query is available in the **Saved Queries** tab of the CMDB Query Builder on the instance.

- [Export and import a query as an update set](#)

Export a saved query definition to an XML file as an update set, which you can later import.

Export a saved query definition to an XML file as an update set, which you can later import.

Before you begin

Role required: To export — cmdb_query_builder (contained in itil and asset). To import — user with permission to import an update set.

Domain in the exported query must be visible in both, source and destination instances.

About this task

Export a query definition as an update set which you can later import and commit. This process lets you port a query between instances, such as from a development environment to a production environment. For more information about exporting and then committing update sets using XML files, see [Save an update set as a local XML file](#).

When exporting a combination query, the integrated Service Mapping query definition is included in the exported update set.

Procedure

1. Export a saved query:
 - a. In the **Filter navigator**, enter qb_saved_query.list and hit Enter to navigate to the Saved Queries table.
 - b. In the Saved Queries list view, select the query that you want to export.
 - c. Click **Actions on selected rows** and then select **Export query**.
 - d. Wait for the **Query Exporter** to complete the export.
2. Import the exported saved query:
 - a. Navigate to **System Update Sets > Retrieved Update Sets**.
 - b. On the Retrieved Update Sets form, click **Import Update Sets from XML**.
 - c. On the Import XML page, click **Choose file** and select the exported XML file. Then click **Upload**.
 - d. Open the new record that was added to the Retrieved Update Sets list view.
 - e. On the Retrieved Update Set form, click **Preview Update Set Batch** and then close the Batch Update Set Preview dialog box.

- f. Click **Commit Update Set Batch** and then close the Commit Update Set Batch dialog box.

Result

The imported query is added to the Query Builder saved queries on the instance.

Settings for CMDB Query Builder

Use settings to control some aspects of the CMDB Query Builder behavior.

Using the Query Builder requires that the [Core UI plugin](#) (com.glide.ui.ui16) is activated.

Open the **Query Builder Settings** dialog box:

1. Navigate to **All > Configuration > CMDB Query Builder**.



2. On the **CMDB Query Builder** page, click the (**Settings**) icon.
3. Click the **Settings** icon again to close the dialog box.

Setting	Description
Display Relationships in Results	Display the relationship between Cls in the query results.
Display Suggested Connections	Filter the CMDB classes and the non-CMDB tables lists in the left pane to display only classes and tables that the selected node on the canvas has a relationship with. You can then drag any item from the filtered list to the canvas, and connect it to the selected node on the canvas.

Setting	Description
	This setting applies only to CMDB queries.
Display Results in New Tab	Display query results in a separate browser tab titled Query Results .

Properties for CMDB Query Builder

Use the CMDB Query Builder properties to configure query processing.

These properties are available for CMDB Query Builder. To view and edit these properties, the admin role is required.

Note: To open the System Properties [sys_properties] table, enter sys_properties.list in the navigation filter.

Properties for CMDB Query Builder

Property	Description
glide.cmdb.query.max_results_limit	Limits the number of results for a scheduled query and in the results section in the Query Builder when you click Load All Results . • Type: integer • Default value: 10000 • Location: Configuration > CMDB Properties > Query Builder Properties
glide.cmdb.query.batch_time_limit_in_sec	Time limit (in seconds) for running one batch to get one batch of query results (100 results). • Type: integer • Default value: 300 • Location: Configuration > CMDB Properties > Query Builder Properties

Property	Description
<p>Time limit (in seconds) for running an entire query to get all results.</p> <p>glide.cmdb.query.query_time_limit_in_sec</p>	<ul style="list-style-type: none"> Type: integer Default value: 1800 Location: Configuration > CMDB Properties > Query Builder Properties
<p>Blacklist of non-CMDB tables that appear in the CMDB Query Builder when creating a CMDB query.</p> <p>glide.cmdb.query.non_cmdb.blacklisted_tables</p>	<ul style="list-style-type: none"> Type: string Default value: empty Other values: Comma separated list of table names (not labels). Can include '*abc' to exclude all tables containing 'abc' in their table name. Location: Configuration > CMDB Properties > Query Builder Properties
<p>glide.cmdb.query.batch_size</p>	<p>Batch size allocated globally when saved queries run.</p> <ul style="list-style-type: none"> Type: integer Default value: 100 Location: Add to System Properties [sys_properties] table.

Monitor system foundations in the CSDM and the CMDB Data Foundations Dashboards (2.2.1)

The ServiceNow® CSDM and CMDB Data Foundations Dashboards store app contains dashboards which provide insights into the key foundational metrics of your CMDB and Common Service Data Model (CSDM). This app provides recommendations to ensure that the CMDB and CSDM are properly configured for optimal usage and to mitigate any potential risks.

Request apps on the Store

Visit the [ServiceNow Store](#) website to view all the available apps and for information about submitting requests to the store. For cumulative release notes information for all released apps, see the [ServiceNow Store version history release notes](#).

The CSDM and CMDB Data Foundations Dashboards store app provides dashboards that complement each other.

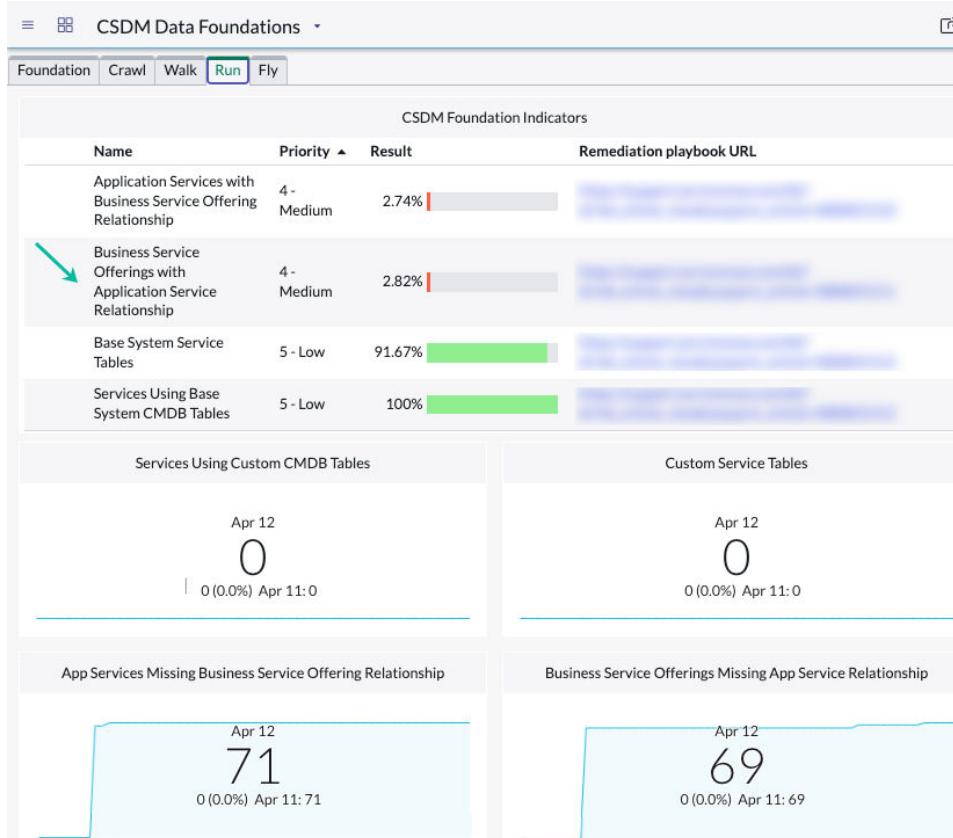
CMDB Data Foundations dashboard

Evaluates various configurations and customizations in the CMDB. This dashboard checks that important data is valid and properly configured, and identifies and provides visibility into potential risks in the implementation. Use the CMDB Data Foundations dashboard to prevent issues and support continuous effective functioning of the CMDB.

CSDM Data Foundations dashboard

The CSDM Data Foundations dashboard displays key CSDM metrics on a single page to assist you in getting the full benefit from your Now Platform® products.

The tabs on the dashboard enable you to select your organization's CSDM implementation stage (foundation, crawl, walk, run, and fly). As a result, the reports on the page show the metrics that are appropriate for the maturity of your CMDB data. In this example, a report on the **Run** tab indicates that several business service offerings don't have the required relationships to application services. With this knowledge, Service owners can add the relationships to ensure that customer service agents get complete information on the upstream impacts of down applications.



The CSDM and the CMDB Data Foundations dashboards capability is provided by the com.snc.cmdb.getwell plugin, which is activated by default in base systems.

For an introduction, watch the ServiceNow [Data Foundations Dashboards for CSDM and CMDB](#) video.

Access the dashboards

To access these dashboards, navigate to **All > Configuration**, and then select **CMDB Data Foundations Dashboard** or **CSDM Data Foundations Dashboard**. You can toggle between the two dashboards by clicking the

change dashboard icon (▼) on the title bar of either dashboard.

Manage performance

Starting with CSDM and the CMDB Data Foundations Dashboards v2.2 you can deactivate CSDM or CMDB metrics.

To improve performance, deactivate metrics that aren't needed or that require extensive resources and affect performance. The active/non-active settings for metrics are preserved across family release upgrades.

To deactivate a metric, you must access the CMDB/CSDM Get Well Metrics [sn_getwell_metric] table where all CMDB and CSDM metrics are stored. Navigate to the list view of the table [sn_getwell_metric]. Then, locate the metric that you want to deactivate and set its **Active** column to **false**. Tiles on the dashboards that are associated with inactive metrics stop collecting data and aren't refreshed.

For more information about deactivating metrics, see [How to "turn off" metric calculations on CMDB and CSDM Data Foundations Dashboards \[KB1430455\]](#).

CMDB Data Foundations dashboard

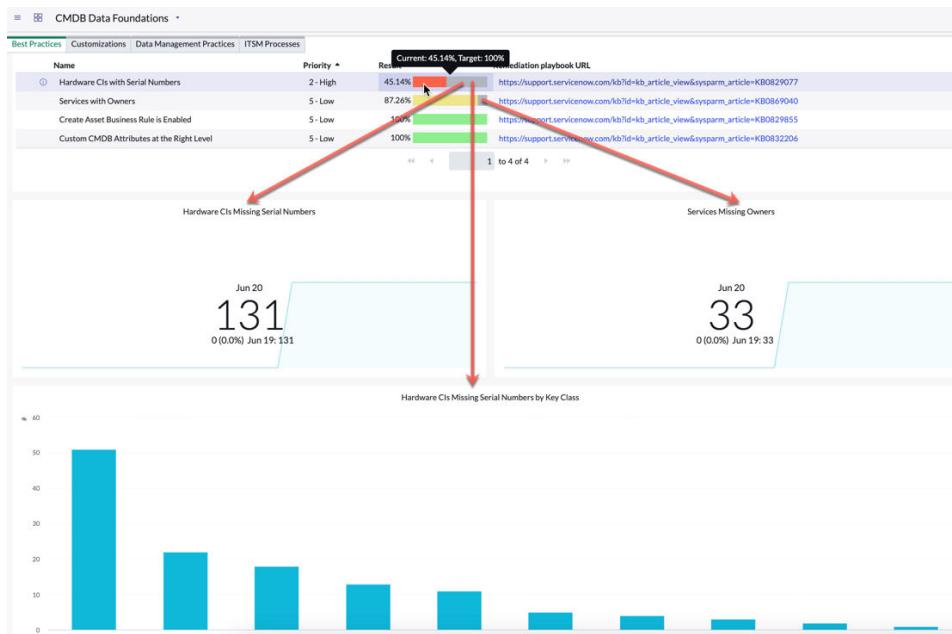
Use the CMDB Data Foundations dashboard to monitor foundational key health-related metrics in CMDB.

Requirements

- To access the CMDB Data Foundations dashboard, users must be configured with the admin, itil_admin, or asset roles. Some metrics that let you drill-down to Performance Analytics (PA) data, might require additional roles for accessing specific tables. Those roles are specified as appropriate per metric.
- The CMDB Data Foundations dashboard adds the following scheduled jobs, which must be running:
 - CMDB Get Well Metric Collection: Calculates and stores details for compliant CIs associated with metrics, populating the list view of the metrics. Data appears on the dashboard only after the first run of this scheduled job. Metrics' scores are stored in the CMDB Data Foundations Metric Scores [sn_getwell_cmdb_score] table. This non-PA job runs daily by default.

- CMDB Data Foundations PA Metric Collection: Calculates the total non-compliant CMDB classes associated with metrics and populates the PA widgets on the dashboard. Also, provides trending data over time for the non-compliant CMDB classes associated with metrics. This PA job runs daily by default.

Overview



The CMDB Data Foundations dashboard provides four tabs, each grouping a distinctive set of CMDB metrics:

- Best Practices:** Checks if usage of tables and properties is as intended for supporting the health of the CMDB. This set of metrics checks adherence of CSDM-related standards, for example, for populating services and relationships.
- Customizations:** Checks how customizations to the CMDB are used. This set of metrics checks that customizations are not used excessively, and that they are used only when needed and are correctly applied.
- Data Management Practices:** Checks if importing third-party data into the CMDB is properly done without compromising the integrity of the CMDB. This set of metrics checks if the imported data is

properly configured and formatted to provide the foundation for CMDB functions.

- **ITSM Processes:** Check if ITSM processes leverage CMDB data.

Columns in the list view of metrics

The following columns appear in the list view of metrics:

Name

Name of metric.

Priority

Priority of an metric. The priority for an metric is a calculation of the weight of the metric and the severity of the percentage score. Priority ranges from **1 - Critical** as the highest priority, to the lowest priority which is **5 - Low**. Metrics are listed in their priority order, starting with the highest priority metric (lowest priority number).

Result

Percentage number of CIs (or the measured item) which are in compliance with the metric. A bar shows the percentage of CIs (or the measured item) which are in compliance versus the percentage of items which are not in compliance. The portion on the bar that represents CIs (or the measured item) which are in compliance, uses the following color scheme to note the level of compliance:

- Red: 0–50% of the total CIs (or the measured item), are in compliance for the metric.
- Yellow: 50–90% of the total CIs (or the measured item), are in compliance for the metric.
- Green: Above 90% of the total CIs (or the measured item), are in compliance for the metric.

The portion on the bar that represents the percentage of items which are not in compliance for the metric, always appears in darker shade. Also, all metric percentage scores are aligned so that a higher percentage score and a smaller darker-shade portion on an metric bar, always indicate the more optimal state.

Note: Some metrics use a lower threshold value which impacts the result that appears in the dashboard. For these metrics, if the percentage of compliant CIs is less than the lower threshold value, then the result appears as 0 for the metric.

Remediation playbook URL

Links to remediation playbook articles in Now Support. These knowledge articles provide context for the respective issue, guidelines to help remediate the issue, and other necessary details to bring CIs into compliance. Click the link and use your Now Support credentials to access the knowledge article.

Performance Analytics widgets

Most metrics are associated with [Performance Analytics \(PA\) widgets](#) (tiles) that provide further details about CIs that are not in compliance with the metric, correlating to the dark-shade portion of the metric bar. Drill down the tiles to access PA widgets, which are provided by the CMDB Data Foundations PA Metric Collection scheduled job.

On the Analytics Hub page:

- Click the **Breakdowns** context menu to see any available breakdowns.



- Ensure that the **Real-time** option is selected () and then click **Show Records** to see a list view of the respective CIs.

Best Practices metrics

The **Best Practices** tab contains the following metrics:

Create Asset Business Rule is Enabled

Checks for the existence of the Create Asset on insert business rule. If the business rule exists, the percentage score for the metric is 100. Otherwise, the percentage score is set to 0.

Custom CMDB Attributes at the Right Level

Percentage of custom attributes that should be added at a higher level in the CMDB hierarchy. For example, the custom attribute Warranty duration was added three times in the Computer class, instead of adding the attribute once at a higher level in the hierarchy, which is more efficient.

- If (total number of attributes that can be moved up > 10): Percentage is set to 0.
- Otherwise: Percentage is set to $(10 - \text{total number of attributes that can be moved up}) * 10$.

Hardware CIs with Serial Numbers

Percentage of Computer or Network Gear CIs with a serial number versus those CIs without. The metric is based on the following conditions:

- Class is an instance of Computer or Gear
- Status = installed
- Operational status = operational
- Serial number is empty

The complete set of CIs for this metric consists of the CIs that satisfy the first three conditions. The CIs that satisfy all conditions are counted as non-compliant Computer or Network Gear CIs, which are without a serial number.

Note: If the percentage of compliant CIs is less than 70%, then the percentage score for this metric is set to 0.

The Hardware CIs Missing Serial Numbers tile shows the total number of CIs which are not in compliance with the metric, correlating the darker-shaded portion of the metric bar. Click the tile to drill down the PA widget for further details, such as the list all those CIs.

The Hardware CIs Missing Serial Number by Key Classes chart shows those CIs that are missing a Serial Number, grouped by key classes. Point to a class bar on the chart to show more details, and click a class bar to access a list view of those CIs for class.

Services with Owners

Percentage of services in which the owned_by field is populated versus those services in which this field isn't populated.

The complete set of CIs for this metric consists of all the services in the Service [cmdb_ci_service] table. The percentage score for this metric is calculated as (Compliant CIs/Complete set of CIs) *100.

The Services Missing Owners tile shows the total number of CIs which are not in compliance with the metric, correlating the darker-shaded portion of the metric bar. Click the tile to drill down the PA widget for further details, such as the list all those CIs.

Customizations metrics

The **Customizations** tab contains the following metrics:

Base System Business Application Table Usage

Checks for the existence of a custom business application table, which is any table that extends [cmdb_ci], whose name starts with "u_" or "x_" and which contains the strings "bus" and "app".

If such custom table exists, then is 0, and if not, the percentage is 100.

The Custom Business Application Tables tile shows any custom business application tables. Click the tile to drill down the PA widget for further details, such as the list all those tables.

Base System CMDB Relationship Types Usage

Percentage of base system relationship types versus custom relationship types that were created by users. Creating custom relationship types can interfere with the integrity of the CMDB therefore this metric can help you prevent such problems.

This metric counts records in the Relationship Type [cmdb_rel_type] table where sys_package is "global" and name is not "Manages::Managed by". The percentage score for this metric is calculated as follows:

- If count is <= 0, then percentage is 100
- If count is > 10, then percentage is 0

- Otherwise percentage = $(10 - \text{count}) * 10$

The Custom CMDB Relationship Types tile shows the relationships which are not in compliance with the metric, correlating the darker-shaded portion of the metric bar. Those are the relationships that were not included in the base system and were custom created. Click the tile to drill down the PA widget for further details, such as the list all those relationships.

Base System Relationship Types Not Deleted or Recreated

Percentage of relationship types that exist in the base system and which were not deleted or recreated, versus those that were deleted and maybe recreated. Deleting and recreating base system relationships can interfere with the integrity of the CMDB, therefore this metric can help you prevent potential problems.

This metric counts the number of base system relationships that were deleted and possibly recreated. The percentage score for the metric is calculated as follows:

- If count is ≤ 0 then the percentage is set to 100
 - If count is > 10 then the percentage is set to 0
 - Otherwise, the percentage = $(10 - \text{count}) * 10$
- .

The Base System CMDB Relationship Types Deleted and/or Recreated tile shows the relationships which are not in compliance with the metric, correlating the darker-shaded portion of the metric bar. Click the tile to drill down the PA widget for further details, such as the list all those relationships.

Custom CMDB Tables Using Standard Naming

Percentage of custom CMDB tables whose name doesn't start with the standard string "u_cmdb_ci". The metric counts tables in the Tables [sys_db_object] table that extend from the CMDB table and whose name starts with the string "u_" (indicating that it is a custom table) but not with "u_cmdb_ci".

The percentage score for this metric is calculated as follows:

- If count <= 0 then percent is 100
- If count > 6 then percent is 0
- If count is between 0 and 6, then percent is $(1-(\text{count}/6)) * 100$

The Custom CMDB Tables Not Using Standard Naming tile shows the tables which are not in compliance with the metric, correlating the darker-shaded portion of the metric bar. Click the tile to drill down the PA widget for further details, such as the list all those tables.

Use of Custom Attributes

Percent of custom attributes that were added to CMDB tables in the base system. Custom attributes are identified by a table name that starts with "cmdb_ci" and a column name that starts with "u_."

The percentage score for the metric is calculated as follows:

- If (total custom attribute count > 50): Percentage is set to 0.
- If (total custom attribute count < 10): Percentage is set to 100.
- If (total custom attribute count is 10–50): Percentage is set to $(\text{total custom attribute count} - 10) / (50 - 10) * 100$.

Data Management Practices metrics

The **Data Management Practices** tab contains the following metrics:

Active CIs Updated in Last 90 days

Percentage of Hardware [cmdb_ci_hardware] or VMware Virtual Machine Instance [cmdb_ci_vmware_instance] active CIs that were updated in the past 90 days versus those CIs that were not (stale CIs). The metric is based on the following conditions:

- Class is an instance of Hardware or VMware Virtual Machine Instance
- Status = installed
- Operational status = operational
- Updated ≥ 90 days

The complete set of CIs for this metric consists of CIs that satisfy the first three conditions. The CIs that satisfy all conditions, are counted as stale

Cl's of the Hardware [cmdb_ci_hardware] and VMware Virtual Machine Instance [cmdb_ci_vmware_instance] classes.

Note: If the percentage of compliant Cl's is less than 65%, then the result displayed for this metric is set to 0.

The Active Cl's Not Updated in 90 Days tile shows the total number of Cl's which are not in compliance with the metric, correlating the darker-shaded portion of the metric bar. Click the tile to drill down the PA widget for further details, such as the list all those Cl's.

Cl's Processed via IRE

Percentage of Hardware [cmdb_ci_hardware] or VMware Virtual Machine Instance [cmdb_ci_vmware_instance] Cl's processed via IRE versus those Cl's not processed via IRE. The metric is based on the following conditions:

- Status = installed
- Operational status = operational
- Sys_id is IN (target_sys_id FROM sys_object_source table)

The complete set of Cl's for this metric consists of the Cl's that satisfy the first three conditions. The Cl's that satisfy all conditions are counted as Hardware [cmdb_ci_hardware] or VMware Virtual Machine Instance [cmdb_ci_vmware_instance] Cl's not processed via IRE.

Note: Viewing all the data provided by PA for this metric, requires the additional role of admin or discovery_admin.

Cl's with Names

Percentage of Hardware, VMware Virtual Machine Instance, or Application Cl's with a name up to 200. The metric is based on the following conditions:

- Class is an instance of Hardware [cmdb_ci_hardware], VMware Virtual Machine Instance [cmdb_ci_vmware_instance], or Application [cmdb_ci_appl]
- Status = installed
- Operational status = operational

- Name is empty

Non-compliant CIs are counted as the CIs that are missing a name, which are those CIs that satisfy all conditions.

The percentage score for the metric is calculated as follows:

- If count of CIs without names is > 200, percent is 0
- If count of CIs without names is < 200, percent is $(200 - (\text{count of CIs without names}) / 200) * 100$

The CIs Missing Names tile shows the total number of CIs which are not in compliance with the metric, correlating the darker-shaded portion of the metric bar. Click the tile to drill down the PA widget for further details, such as the list all those CIs.

CIs with Relationships to Parent and Child

Percentage of non-orphan CIs versus orphan CIs.

The complete set of CIs for this metric consists of CIs in records from the CI Relationship [cmdb_rel_ci] table. CIs in which [parent.sys_class_name=(empty)] OR [child.sys_class_name=(empty)], are counted as orphan CIs.

The Orphan CIs tile shows the total number of orphan CIs, correlating the darker portion of the metric bar. Click the tile to drill down the PA widget for further details, such as the list all those CIs.

Handled Duplicate CIs

Percentage of duplicate CIs of the Hardware or VMware Virtual Machine Instance classes, that were remediated, up to 200. If the number of unhandled duplicate CIs is greater than the upper threshold of 200, then is set to 0. The metric is based on the following conditions:

- Class is an instance of the Hardware [cmdb_ci_hardware] or the VMware Virtual Machine Instance [cmdb_ci_vmware_instance] class.
- Status = installed
- Operational status = operational
- Duplicate_of is not empty

Non-compliant CIs are counted as those CIs that satisfy all conditions. Non-compliant CIs are those Hardware [cmdb_ci_hardware] or VMware Virtual Machine Instance [cmdb_ci_vmware_instance] duplicate CIs that were not remediated. For more information, see [Duplicate CIs](#).

The percentage score for this metric is calculated as follows:

- If count of non-compliant CIs is > 200 then percent is 0
- If count of non-compliant CIs is < 200 then percent is $(200 - \text{count of non-compliant CIs}) / 200 * 100$

The Unhandled Duplicate CIs tile shows the total number of CIs which are not in compliance with the metric, correlating the darker-shaded portion of the metric bar. Click the tile to drill down the PA widget for further details, such as the list all those CIs.

Installed Server CI Naming Reflecting Hostname

Percent of Server CIs where name reflects the host name, up to the first period in the name. The complete set of CIs for the metric are all the servers in the Server [cmdb_ci_server] table in which install_status=1 and host_name is not empty.

Examples of compliant CIs:

- Server Name is “abc”, and host name is “abc.1.2.3”
- Server Name and host name are the same

The percentage score for the metric is calculated as $(\text{Compliant CIs} / \text{Total CIs}) * 100$.

The Installed Server CI Naming Not Reflecting Hostname tile shows the number of Server CIs which are not in compliance with the metric. Click the tile to drill down the PA widget for further details, such as the list all those CIs.

Note: If you need to manually refresh the Installed Server CI Naming Not Reflecting Hostname widget, then you must first run the CMDB Get Well Metric Collection job, and then run the CMDB Data Foundations PA Metric Collection job.

Managed CIs with Model Entries

Percentage of Hardware [cmdb_ci.hardware] CIs with a model ID versus those CIs without.

The complete set of CIs for this metric consists of all the CIs in the Hardware [cmdb_ci.hardware] table. Compliant CIs are those in which model_id is populated.

The Managed CIs Missing Model Entries tile shows the total number of CIs which are not in compliance with the metric, correlating the darker-shaded portion of the metric bar. Click the tile to drill down the PA widget for further details, such as the list all those CIs.

Servers with Location

Percentage of Server CIs in which Location is populated versus Server CIs in which Location isn't populated. The complete set of CIs for this metric are the Server CIs in the Server [cmdb_ci_server] table in which install_status =1 and operational_status=1. Compliant CIs are those CIs in which also location is not null.

The percentage score for the metric is calculated as (Compliant CIs/Total CIs)*100.

The Servers Missing Location tile shows the number of Server CIs which are not in compliance with the metric. Those are the Server CIs in which Location is empty. Click the tile to drill down the PA widget for further details, such as the list all those CIs.

Unique Locations

Percentage of unique locations versus non-unique (duplicate) locations.

The complete set of locations for this metric consists of all the locations in the Location [cmn_location] table records, in which name is not null. A duplicate location is counted as a record in which name is not null, and the same name is used more than once.

The percentage score for the metric is calculated as (Compliant CIs/Total CIs)*100.

The Duplicate Locations tile shows the total number of CIs which are not in compliance with the metric, correlating the darker-shaded portion of

the metric bar. Click the tile to drill down the PA widget for further details, such as the list all those CIs.

Note: If you need to manually refresh the Installed Server CI Naming Not Reflecting Hostname widget, then you must first run the CMDB Get Well Metric Collection job, and then run the CMDB Data Foundations PA Metric Collection job.

ITSM Processes metrics

The **ITSM Processes** tab contains the following metrics:

Changes Referencing a CI

Percentage of change requests with a reference to CIs versus those change requests without. The complete set includes all change requests created in the last 90 days. Compliant records are those change requests in which the cmdb_ci field is not empty.

The percentage score for the metric is calculated as (Compliant change requests/Total change requests) * 100.

The Changes Not Referencing a CI tile shows the total number of records which are not in compliance with the metric, correlating the darker-shaded portion of the metric bar. Click the tile to drill down the PA widget for further details, such as the list all those change requests.

Note: Viewing all the data provided by Performance Analytics for this metric (when drilling down), requires the additional role of itil or sn_change_read.

Changes Relating to both a Service and a CI

Percentage of changes relating services to CIs versus those changes that don't. The complete set of CIs for this metric consists of all change request records. Compliant records are those change requests in which both, the cmdb_ci and the business_service fields, are not empty.

The Incidents Relating to Neither a Service nor a CI tile shows the total number of records which are not in compliance with the metric, correlating the darker-shaded portion of the metric bar. Click the tile to drill down the PA widget for further details, such as the list all those CIs.

Note: Viewing all the data provided by Performance Analytics for this metric (when drilling down), requires the additional role of itil or sn_change_read. Click the tile to drill down the PA widget for further details, such as the list all those CIs.

Incidents Referencing a CI

Percentage of incidents referencing a CI versus those incidents that don't. The complete set of records includes all incidents created in the last 60 days. Compliant incidents are all those in which the cmdb_ci field is not empty.

The percentage score for the metric is calculated as (Compliant Incidents/Total Incidents)*100.

The Incidents Not Referencing a CI shows the total number of incidents which are not in compliance with the metric, correlating the darker-shaded portion of the metric bar. Click the tile to drill down the PA widget for further details, such as the list all those incident records.

Note: Viewing all the data provided by Performance Analytics for this metric (when drilling down), requires the additional role of itil or sn_incident_read.

Incidents Relating to both a Service and a CI

Percentage of incidents relating to both a service and a CI versus those incidents that don't. The complete set for this metric is all incident created in the last 60 days. Compliant records are those incidents in which both, the cmdb_ci and the business_service fields, are not empty.

The percentage score for the metric is calculated as (Compliant Incidents/Total Incidents) * 100.

The Incidents Relating to Neither a Service nor a CI shows the total number of records which are not in compliance with the metric, correlating the darker-shaded portion of the metric bar. Click the tile to drill down the PA widget for further details, such as the list all those incident records.

Note: Viewing all the data provided by Performance Analytics for this metric (when drilling down), requires the additional role of itil or sn_incident_read. Click the tile to drill down the PA widget for further details, such as the list all those Cls.

Configure the CSDM Data Foundations dashboard

Use the CSDM Data Foundations dashboard to monitor and evaluate key foundational metrics of the CSDM framework.

Before you begin

For an introduction to the dashboard, see [Viewing the CSDM Data Foundations dashboard](#).

- Before you use the dashboard for the first time, populate the CSDM metrics: Navigate to **All > System Scheduler > Scheduled Jobs** and run the CSDM Get Well Metric Collection job.
- The CSDM Data Foundations dashboard adds the following scheduled jobs that must be running:
 - CSDM Get Well Metric Collection: Calculates and stores details for compliant Cls associated with metrics. Data appears on the dashboard only after the first run of this scheduled job. metrics' scores are stored in the CSDM Data Foundations Metric Scores [sn_getwell_csdm_score] table. The job runs daily by default.
 - CSDM Data Foundations PA Metric Collection: Calculates the total count of non-compliant Cls that are associated with each metric. It also provides trending data over time for the non-compliant Cls associated with metrics.
- Role required: app_service_admin, app_service_user, asset, cmdb_read, itil_admin, portfolio_admin, service_viewer, or technology_service_owner

Procedure

1. Navigate to **All > CSDM > Configuration > CSDM Data Foundations Dashboard**.
2. Select a tab.

The tabs on the dashboard enable you to select your organization's CSDM implementation stage (foundation, crawl, walk, run, and fly). As a result, the reports on each tab display the metrics that are appropriate for the maturity of your CMDB data.

3. Review the reports.

Note the percentages and color-coding in the **Result** column for each metric.

- If the percentage is 100%, the CSDM framework has the information it needs. You don't need to do anything else.
- Otherwise, required information is missing and additional actions are required. Continue with [step 5](#).

Note the metrics on the **Foundation** tab:

Named Product Models with Product Owners

Shows cmdb_model records that meet the following conditions:

- **Status** = in production
- **Name** and **Owner** is not empty

Configuration Item Status Values

Shows the percentage of CIs with default status values.

- 80%: At least 1 to 5 CIs have custom status values.
- 60%: 6 to 10 CIs have custom status values.
- 40%: 11 to 15 CIs have custom status values.
- 20%: 16 or more CIs have custom status values.

To view the default base-system status values, enter `sn_getwell_oob_status_table_field.list` in the navigator **Filter** text box. The Configuration Item Status Values form displays the list of elements and associated tables. Select a table name to see the list of default labels and values.

		Configuration Item Status Values	Search	Label	Search
		Table field = cmdb_ci			
		Label		Value	
<input type="checkbox"/>		<u>Retired</u>		7	
<input type="checkbox"/>		<u>Pending Install</u>		4	
<input type="checkbox"/>		<u>Installed</u>		1	
<input type="checkbox"/>		<u>Stolen</u>		8	
<input type="checkbox"/>		<u>Absent</u>		100	
<input type="checkbox"/>		<u>In Stock</u>		6	
<input type="checkbox"/>		<u>Pending Repair</u>		5	
<input type="checkbox"/>		<u>On Order</u>		2	
<input type="checkbox"/>		<u>In Maintenance</u>		3	

Business Units with Companies

Shows business unit records where the **Company** field is not empty.

Locations with Parents.

Shows cmdb_ci records that meet the following conditions:

- **Status** = installed
- **Operational status** = operational
- **Location** and **Location.parent** is not empty

4. Select the tiles associated with the foundational metrics to access Performance Analytics widgets.

Performance Analytics widgets are provided by the CSDM PA Metric Collection scheduled job. These widgets provide trending data over time for the non-compliant CIs associated with the metric.



- Ensure that the real-time option is selected () and then select **Show Records** to view the list of CIs.
- Select the **Breakdowns** context menu to view available breakdowns.

5. Scroll to the list of CIs in the **Custom Status Values** related list.

The charts show the number of custom values that have been defined for each element. Click a chart to view custom values that have been defined for the element. This example shows the custom label-value combination for the **Absent** status.

The screenshot shows a table with columns: Element, Label, and Value. The first two rows are highlighted with a red border. The first row contains 'cmdb_ci' under Element, 'Absent2' under Label, and '100' under Value. The second row contains 'cmdb_ci' under Element, 'Absent' under Label, and '101' under Value. The third row contains 'service_offering' under Element, 'Installed by me' under Label, and '1' under Value.

Element	Label	Value
cmdb_ci	Absent2	100
cmdb_ci	Absent	101
service_offering	Installed by me	1

6. Select a CI to drill down to the form view.

The form view provides the required information. If you don't see the form, you may not have sufficient access privileges. Contact your ServiceNow administrator.

7. When you're finished using the form, select **Update** or **Delete** to return to the list view.
8. Navigate to return to the CSDM Data Foundations dashboard.

Result

The key foundational metric results are available for you to review and analyze.

CMDB Health

Monitoring and maintaining the health of the CMDB is essential to an effective and continuous use of the product. Health indicators such as duplicate CIs, required CI fields, and audits contribute to the calculation of health scorecards at the CI, class, and CMDB level.

Note: CMDB Health doesn't support non-CMDB tables.

The health of the CMDB data is monitored and reported for the following KPIs, each further consisting of sub metrics:

- Completeness: CIs are tested for required and recommended fields that are not populated.
- Correctness: CIs are tested against pre-defined data integrity rules such as identification rules (to detect duplicate CIs), orphan CI rules, and stale CI rules.
- Compliance: The CMDB data is audited for adherence to pre-defined certificates.
- Relationships: The health of CI relationships is tested for indicators such as orphan and duplicate relationships. And for compliance with suggested relationships, hosting and containment rules.

After CIs are tested for various health indicators, the results are aggregated at the class level, and eventually at the overall CMDB level. You can configure how health is calculated and the weight of each KPI

and each metric at every level of the aggregation. For most health tests, you can configure the health tests themselves.

CMDB Health experience in CMDB Workspace

You can use the [CMDB Workspace](#) landing page and its views to access CMDB health details that are based on CMDB Health activities and aggregations. For example:

- Use the Important actions tile on the landing page to access cards with CMDB Health-related tasks such as de-duplication tasks.
- Use the CMDB Health tile to see overall health metrics for CIs and relationships, to navigate to the CMDB Health and CMDB Relationship Health dashboards, and to drill down to more health details for specific CIs.
- Use tiles throughout the CMDB Workspace to drill down to health overviews of specific CIs.

CI remediation

CMDB Health provides a framework for configuring [CI remediation](#). Remediation lets you proactively apply corrective actions to unhealthy CIs in a managed and standardized fashion.

Domain separation

CMDB Health is domain aware. If the domain separation plugin has been activated, then the CMDB dashboard displays health based on data, rules, and settings from the logged-on user domain. If rules and settings are not defined for a child domain, then the parent's settings are applied, recursively.

Metric tests from the global domain propagate to subdomains. However, subdomains can have their own local metric tests which override the global domain tests. Up until the San Diego release, subdomain local metric tests were applied to the subdomain CIs and also to the global domain CIs (which are visible on subdomains). Global domain CIs that failed metric tests of local subdomains, could have generated large amounts of data due to duplicated data.

Starting with the Tokyo release, CIs in the global domain are evaluated only against metric tests specified in the global domain. In subdomains,

local metric tests are applied only to the CIs in that subdomain and are not applied to the global domain CIs (even though the global domain CIs are visible in the subdomain). Health results for CIs in the global domain appear on subdomains and health results on subdomains reflect this new behavior.

See [Domain separation in CMDB Health](#) for more information.

Setup

To start gathering and aggregating health data, you need to enable the CMDB Health-related jobs (CMDB Health Dashboard jobs) which are initially disabled. You also need to configure CMDB Health related system properties and health KPI and metric test rules, to customize how aggregated data is calculated and other CMDB Health behavior.

For all the details about setting up and configuring CMDB Health, see [Setup and configure CMDB Health](#).

- [Domain separation in CMDB Health](#)

This is an overview of domain separation as it pertains to CMDB Health. Domain separation enables you to separate data, processes, and administrative tasks into logical groupings called domains. You can control several aspects of this separation, including which users can see and access data.

- [CMDB Health KPIs and metrics](#)

The overall CMDB health score consists of three Key Performance Indicators (KPIs) which are correctness, compliance, and completeness, each further consisting of sub-metrics. Each KPI and metric is associated with a scorecard that determines its contribution to the aggregated health at the overall CMDB level, class, and CI level.

- [CMDB Health dashboards](#)

CMDB dashboards display CMDB health reports and let you configure the CMDB health KPIs and metrics that CIs are evaluated for.

- [View CMDB health reports](#)

The CMDB dashboard serves as a central location to view aggregated health reports for your CMDB at a glance which helps you understand

the CMDB health status. Also, it provides functions to address health issues, and improve CMDB health.

- [View services health reports](#)

The CMDB service dashboard serves as a central location to view aggregated health reports for services at a glance. Also, it lets you drill into a service to perform remediation actions that address health issues, and that improve CMDB health. The CMDB service dashboard uses the Performance Analytics framework for dashboards and employs the capabilities it provides.

- [View CMDB groups health reports](#)

The CMDB group view dashboard serves as a central location to view aggregated health reports for CMDB groups at a glance. Also, it lets you drill into a CMDB group to perform remediation actions that address health issues, and that improve CMDB health. The CMDB group view dashboard uses the Performance Analytics framework for dashboards and employs the capabilities it provides.

- [View CI health](#)

The CI dashboard is a central location displaying health report for an individual CI, history of changes to the CI in a timeline view, and the relation formatter. The CI dashboard also displays incidents, changes, and other tasks affecting the CI, and business services affected by the CI. You can access the CI dashboard from a CI form, or from the CMDB dashboard.

- [View CI relationships health](#)

View aggregated orphan, stale, and duplicate CI relationships in the CMDB dashboard. You can configure the relationship scorecards, but you cannot configure the underlying relationship KPI health tests.

- [Create CMDB remediation rule](#)

A CMDB remediation rule is associated with a task that was created for a failed CMDB health test. A CMDB remediation rule is applied automatically or manually to execute a remediation workflow that can, for example, delete stale CIs.

- [CMDB Health process tracking](#)

Use the following information to track and resolve issues with the CMDB Health processes.

Domain separation in CMDB Health

This is an overview of domain separation as it pertains to CMDB Health. Domain separation enables you to separate data, processes, and administrative tasks into logical groupings called domains. You can control several aspects of this separation, including which users can see and access data.

Overview

CMDB dashboards should be set up with their own set of rules to best accommodate how the user needs them. CMDB dashboard jobs adhere to those rules to produce reports. These are covered in separate sections below.

How domain separation works in CMDB Health

For dashboards to be the most effective, users should configure the dashboard accordingly. This is done by setting up the orphan, staleness, and inclusion rules to meet their needs, which then affect the reports displayed on the dashboard.

The settings and metrics define different aspects of each application because each domain can be configured differently. These rules are set up in addition to those that are included in the base system. There are different types of owners for different CIs; each domain has its own set of rules.

Metric tests from the global domain propagate to subdomains. However, subdomains can have their own local metric tests which override the global domain tests. Up until the San Diego release, subdomain local metric tests were applied to the subdomain CIs and also to the global domain CIs (which are visible on subdomains). Global domain CIs that failed metric tests of local subdomains, could have generated large amounts of data due to duplicated data.

Starting with the Tokyo release, CIs in the global domain are evaluated only against metric tests specified in the global domain. In subdomains, local metric tests are applied only to the CIs in that subdomain and are not applied to the global domain CIs (even though the global domain

CLs are visible in the subdomain). Health results for CLs in the global domain appear on subdomains and health results on subdomains reflect this new behavior.

Note: Domain separation is on by default, but each domain can be configured as needed.

Health Preferences

Configure these preferences during setup:

1. Global system properties that control CMDB Health – System properties are not domain separated. To learn more see [CMDB Health system properties](#).
2. [CMDB Health Dashboard Jobs](#) – There is a dashboard job for each major KPI, such as Completeness. That job finds the health of the CLs across all the enabled domains. There is only one job run for all domains and jobs themselves are not domain separated.

Users can define the frequency with which they want to run jobs; the report runs for all the domains. The more domains included in the job, the longer the job runs.

3. [Health Metrics](#) – These selections are domain-separated and adhere to the established “system overrides” logic of domain separation. Changes are made according to the domain for which the user is logged in. Base system values are defined at the global domain. The overriding domain logic means these values apply for all domains. If users want different values for a domain, they must be logged in to a specific domain and change the property from there. The new property setting applies only to that domain and any domain that inherits this domain. To learn more, see [Health Metrics](#).

Note: Regarding the Completeness, Compliance, and Correctness KPIs: Users can disable this KPI if they don't want to see that as part of the dashboard score. All these settings are domain-separated and the user can define specific properties for the domain.

- a. Weighted averages – These settings can affect all or part of the metrics in Completeness, Compliance, Correctness, and Relationship. They can be set differently for different domains.

- b. Active – This setting is the most important because it affects how long the jobs run. The more domains with flags set to Active, the longer the jobs take. It's best to select only those domains you wish to be Active and render the rest as Active = **false**. You can set this in Health Preferences. The default settings for global domain are Active = **true**, but you can modify or disable specific domains the user wants to see in the dashboard. Users should consider the domain hierarchy when changing these values. If there is a large number of domains (>100) the job can take a very long time. To mitigate this, set Active to **false** for all the root domains, thereby disabling all the other domains in the hierarchy. If there is a rule at the top, all child domains inherit that rule.
- c. Failure Threshold, Create Task, Task Assignee Group – All these settings can be set differently for different domains depending on what is needed in each domain.
- d. Exceptions – For Relationship metrics (relationship, duplicate relations, orphan relations, stale relations) the failure threshold setting is not domain separated. The Failure Threshold for the global domain is applied to all domains. For example, even if users were to override the Failure Threshold for a domain, the global domain setting for Threshold is still applied.
- e. Troubleshooting / Implementation detail – These settings are stored in the cmdb_health_metric_pref table, which is domain separated.

CMDB Health-related rules

See CMDB Health-related rules settings at:

- Required
- Recommended
- Orphan
- Staleness

Most of the CMDB Health-related rules are domain separated and provided by the users. Users can define different rules for different domains by logging in to each domain and adding/overriding rules in the CI Class Manager.

1. Completeness

- a. Required fields – These are based on the class schema defined in the platform's [System dictionary](#) and is fixed for all domains. These cannot be changed.
- b. Recommended fields – These are domain separated. The table used is `cmdb_recommended_fields`, which is domain separated. The user can set these up for different domains.

2. Correctness

- a. Duplicates – Duplicates are based on Identification rules, which are not domain separated, so the same rules apply to all domains.
- b. Orphan – Orphan rules are domain separated; there are different orphan rules for different domains. The table used is `cmdb_health_orphan_rule` and is domain separated.
- c. Staleness – Staleness rules are domain separated. The table used is `cmdb_health_staleness_rule`. The base system rule (60 days) is set for global domain so is inherited by all domains as the default rule.

3. Compliance

Audit – Audit scores are based on the desired state or scripted audits defined in the compliance module by the user. Audits themselves are domain separated. When audit score evaluation is enabled for a domain, scores become based only on the audits visible in that domain.

Health inclusion rules:

- Health inclusion rules are domain separated. The rules are stored in the `cmdb_health_config` table which is domain separated.
- Each domain can have its specific health inclusion rules and domain-specific rules for each sub-metric.
- When a health inclusion rule is defined globally, all sub-domains inherit the rule according to the domain structure and the rule can be overridden at any domain.

- When a health inclusion rule is defined at the Configuration Item [cmdb_ci] class level, all descending classes inherit the rule and the rule can be overridden at any class level.

Health Dashboards (CMDB View/ Service View / Group View)

In general, CMDB Health dashboards are domain aware and show data according to the logged-on domain user. If a user is logged into a domain and views a health dashboard:

1. Only scores for enabled metrics in that domain display (based on the Health Preferences Active flag as discussed above).
2. All scores are based on CIs that are visible from the specific domain. (These are regular domain visibility rules: From that domain you can see CIs in global domain, the specific domain, any child domain of that domain or any domain that gets directly or indirectly contained by that domain.)
3. The dashboard view is based on domain rules defined in domain mapping, as opposed to those provided by the logged-in user. This view overrides any additional domain visibility rules that a logged-in user might have. The admin sets the basic rules, but does not set each individual domain. The admin can give specific users or user groups additional visibility to other domains and the dashboard still does not change. The dashboard strictly follows the domain rules mentioned above, based on the domain hierarchy for the domain in which the user is logged in.
4. As explained in the Health Preferences section, users can define different preference values for any domain which impact the scores reported in the dashboard. Preferences that can impact scores include **Weighted Averages**, **Failure Threshold**, and **Active**.
5. As explained in the CMDB Health Rules section, the scores reported for the metrics are based on the health rules defined for them (staleness, orphan, recommended, audit, and inclusion rules) which can be defined differently for a specific domain (in the CI Class Manager). Only the required metric and duplicate metric are based on rules that apply in all domains.
6. Service View/ Group View – These reports also largely follow the above points. Typically, these views differ from various views/filters for

the Health Report. One is based off business rules, the other is based off CMDB Health groups.

Related tasks

- [View CI health](#)
- [View CI relationships health](#)
- [Create CMDB remediation rule](#)

Related concepts

- [CMDB Health dashboards](#)
- [View CMDB health reports](#)
- [View services health reports](#)
- [View CMDB groups health reports](#)
- [Domain separation and Configuration Management Database \(CMDB\)](#)

Related reference

- [CMDB Health KPIs and metrics](#)
- [CMDB Health process tracking](#)

Setup and configure CMDB Health

The data collection system is highly configurable, however, the base system is minimally configured for aggregating CMDB health data. Most importantly, the CMDB Health Dashboard jobs are disabled by default and data is not collected. To display valuable and meaningful data, you should review and adjust settings.

1. Review [CMDB Health KPIs and metrics](#) to learn what CMDB Health can monitor, and what needs to be configured to enable and support each metric.
2. For each KPI and associated metric that you want monitored, define the necessary rules and fulfill other needed requirements. For

example, create orphan rules for detecting orphan records, if you are interested in this metric.

3. Review and adjust the threshold ranges for best, at risk, and critical states for the CMDB health metrics scorecards - see [Configure CMDB Health scorecard thresholds](#).
4. Set metric aggregation preferences, deactivate KPI and metrics that you are not interested in reporting, set failure thresholds, and adjust weighted averages of aggregation - see [Configure KPI and metrics aggregation preferences](#).
5. Narrow the scope of CIs that are included in health calculations - see [Create health inclusion rule](#).
6. Enable the Health Dashboard jobs for the KPIs that you want reported - see [Enable and configure a CMDB Health Dashboard job](#).
7. [Customize the CI dashboard](#) (optional).

Configure the following system properties to customize how CMDB Health is monitored and evaluated.

Role required: itil_admin

Note: To open the System Property [sys_properties] table, enter `sys_properties.list` in the navigation filter.

Property	Description
Max time in minutes for which individual metric processor will run in each scheduled cycle [glide.cmdb.health.metricProcessor.r.maxRunningTime]	If processing of a metric exceeds the specified time, CMDB Health processing halts until the next CMDB Health job is scheduled to run. <ul style="list-style-type: none">• Type: integer• Default value: 120

Property	Description
	<ul style="list-style-type: none"> Location: Navigate to All > Configuration > Health Preference. In the right hand-side navigator, click System Properties. <p>For performance reasons, it is recommended not to set this property to a value greater than 120.</p> <p>Note: If you enter an invalid value, the default value is used.</p>
glide.cmdb.logger.use_syslog.CMDBHealth	<p>A comma-separated list that controls the level of logging of CMDB Health jobs. Logging creates entries in the system logs to capture messages generated by the health auditing process each time they run. This logging helps debugging if there is a failure.</p> <p>For example, to log error and info messages, set the value to 'error,info'.</p> <ul style="list-style-type: none"> Type: String Default value: error Other possible values: Comma-separated list with any of the following values: <ul style="list-style-type: none"> info error warn

Property	Description
	<p>Or '*' which is equivalent to including all possible values.</p> <ul style="list-style-type: none"> Location: System Property [sys_properties] table.
glide.cmdb.health.src.cmdb_health_audit_only	<p>When set to true, disables health results from sources other than CMDB Health audit (such as cloud discovery). Only results generated by CMDB Health audit appear in the CMDB dashboard.</p> <p>For example, by default, if a CI is determined to be stale by Discovery, then that CI appears as stale in the CMDB dashboard even though CMDB Health audit did not determine that CI to be stale. To disable these stale CI health results, set the property to true.</p> <ul style="list-style-type: none"> Type: true false Default value: false Location: System Property [sys_properties] table. Learn more: <ul style="list-style-type: none"> CMDB Health KPIs and metrics Discovery for VMware vCenter
glide.cmdb.health.staleness_exclude_dependent_cis	Exclude dependent CIs for the staleness CMDB Health metric.

Property	Description
	<p>When enabled, dependent CIs are not checked for staleness, regardless of any staleness or inclusion rules that are defined for the respective CI types.</p> <ul style="list-style-type: none">• Type: true false• Default value: false• Location: System Property [sys_properties] table.

Enable and configure the jobs that process CMDB health tests, to start calculating CMDB health scores for the completeness, compliance, correctness, and relationship KPI. These health scores are then aggregated into the overall CMDB health report.

Before you begin

Role required: admin

About this task

In the base system, CMDB Health Dashboard jobs are disabled by default. Enable and configure the respective job for the CMDB health KPI that you want data collected and aggregated for. You can schedule a job to run on a recurring schedule, or execute it once at any time.

For more information about how CMDB Health Dashboard jobs work, see the [Understanding the CMDB HealthDashboard Numbers](#) blog post in the ServiceNow Community.

Procedure

1. Navigate to **All > Configuration > CMDB Dashboard > CMDB View**, and then click **CMDB Health Dashboard Jobs**.
2. Select a job that you want to enable or configure.

CMDB Health Dashboard job	Description
CMDB Health Dashboard - Completeness Score Calculation	Script for calculating the completeness KPI of CMDB health.
CMDB Health Dashboard - Compliance Score Calculation	Script for calculating the compliance KPI of CMDB health.
CMDB Health Dashboard - Correctness Score Calculation	Script for calculating the correctness KPI of CMDB health.
CMDB Health Dashboard - Relationship Score Calculation	Script for calculating the CI relationships KPI of CMDB health.
CMDB Health Dashboard - Relationship Compliance Processor	Script for calculating compliance of relationships with suggested relationships, and with hosting and containment rules.

3. Review the default configuration, and update as necessary.

Field	Description
Name	Job name. Leave this field with the pre-populated name.
Active	Select to activate the job.
Conditional	If selected, the scripted condition must evaluate to true before the job can run.
Run	Configure the schedule for job execution, or select On Demand to run the job manually when needed. If Active is selected, then additional fields appear

Field	Description
	according to your choice. For example, Repeat Interval (when Run is set to Periodically), and Time for various settings of Run . Fill in the details to set precise running times.
Time zone	Local time zone.
Run this script	The job's script. Note: Changes to the script might result in unexpected behavior.

4. Click **Execute Now** to run the job once immediately.

Result

After you enable a CMDB Health Dashboard job, the results for the KPI are aggregated and displayed in the CMDB dashboard and CI dashboard, at the CMDB, class, and CI levels.

Configure the thresholds for best, at risk, and critical state definitions for the KPIs and metrics scorecards. You can configure these settings globally for the entire CMDB, or individually per class.

Before you begin

Role required: itil has read access, itil_admin (on top of itil) has full access.

About this task

Scorecard thresholds are used to determine overall metric state, and are defined by upper and lower thresholds. For example, scorecard thresholds for completeness:

- 0 - lower threshold: Best state
- Lower threshold - upper threshold: At risk state

- Upper threshold - 100: Critical state

In the base system, upper thresholds are set to 67 and lower thresholds are set to 33 for all KPIs and metrics. You can adjust scorecard thresholds to reflect the range of failures that should be used for each health state. Applying the change to a scorecard is based on the selected class in the CI Classes list:

- If the top level Configuration Item class is selected in the CI Classes list, changes to metric scorecards apply to the entire hierarchy.
- If any other class is selected in the CI Classes list, changes to metric scorecards apply to the selected class.

For CMDB groups, you can specify a separate set of scorecard thresholds, per CMDB group/KPI or metric. See [Configure CMDB groups scorecard thresholds](#) for more details.

Procedure

1. Navigate to **All > Configuration > CI Class Manager**.
2. Click **Hierarchy** to display the CI Classes list, and then select a class to set scorecards for.
3. In the class navigation bar, expand **Health**.
4. To configure the overall scorecard, select **Other Scorecards** and then select **Overall Scorecard**.
5. To configure a scorecard for any of the KPIs (such as Compliance), or for relationship-related metrics (such as duplicate relations):
 - a. Select **configuration item** at the top of the **Hierarchy** list.
 - b. Click a KPI item, **Completeness**, **Compliance**, or **Correctness**.
 - c. Click the tab **CMDB Completeness Scorecard**, **CMDB Compliance Scorecard**, or **CMDB Correctness Scorecard**.
- When you select the top-level Configuration Item class, changes to threshold settings in any metric scorecard apply to the entire class hierarchy.
6. To configure a scorecard for any metric:

- a. In the CI Classes list, select the class to which the updated scorecard should apply to. Select the top level Configuration Item to apply the change to the entire hierarchy.
 - b. Click the KPI that contains the metric for which you want to configure scorecard. For example, click **Completeness** to configure the Required Fields scorecard.
 - c. Select the scorecard tab of the metric to configure, such as **Required Fields Scorecard**. You might need to click **New** to edit the scorecards.
7. Slide the threshold sliders, or enter specific numbers to increase or to decrease the threshold bars to fit your definitions for best, at risk, and critical levels for the scorecard.
8. Click **Save**.

Related reference

- [CMDB Health KPIs and metrics](#)

Each CMDB group can have its unique set of scorecard thresholds for best, at risk, and critical state definitions for specific KPIs or metrics.

Before you begin

Role required: itil_admin (on top of itil)

About this task

Scorecard thresholds are used to determine overall metric state, and are defined by upper and lower thresholds:

- 0 - lower threshold: Best state
- Lower threshold - upper threshold: At risk state
- Upper threshold - 100: Critical state

In the base system, for all KPIs and metrics, upper thresholds are set to 67 and lower thresholds are set to 33. You can adjust scorecard thresholds for a specific CMDB group per KPI or metric, to reflect the range of failures that should be used in health reporting.

CMDB groups scorecard thresholds are stored in the [cmdb_health_scorecard_group_threshold] table.

Procedure

1. In the search box in the navigation bar, enter `cmdb_health_scorecard_group_threshold.list` and press the Enter key.
2. In the **CMDB Health Group Scorecard Thresholds** list, click **New**.
3. Fill out the form.

Field	Description
Upper threshold	Upper range of percentage of CIs failing the specified metric tests, that is used to calculate best and at risk states.
CMDB Group	The CMDB group to which this scorecard threshold setting applies to.
Lower threshold	Lower range of percentage of CIs failing the specified metric tests, that is used to calculate at risk and critical states.
Metric	The metric to which the scorecard threshold setting applies to.

4. Click **Submit**.

Related tasks

- [Configure CMDB Health scorecard thresholds](#)

Metrics health scorecards are aggregated into their respective KPI, which in return are aggregated into the overall CMDB Health report. Set aggregation preferences for KPIs, and for each of their respective

metrics, deactivate KPIs and metrics that you are not interested in reporting, and adjust weighted averages of aggregation.

Before you begin

To start collecting and reporting CMDB health KPIs and metrics, you must first [enable and configure the CMDB health dashboard jobs](#).

Role required: itil_admin (on top of itil)

About this task

The completeness KPI for example, consists of the metrics required fields and recommended fields, each contributing a different weight to the sum. You can configure the proportional weight of required fields and recommended fields within completeness to be 25 and 75 respectively. You can also configure the proportional weight of completeness, compliance and correctness within the aggregated sum of the overall CMDB health.

Note: Non-active KPI or metrics are displayed on the CMDB dashboard in faded coloring, displaying the most recent aggregations that were calculated when the KPI or metric was active.

If Domain Support - Domain Extensions is activated, then you can configure aggregation preferences per domain.

In the ServiceNow base system, the weights of KPIs have default settings, and metrics are globally set.

Procedure

1. Navigate to **All > Configuration > Health Preferences**.
2. Select **Health Metrics** on the right-hand side navigator.
3. From the **Select Metric** list select one of the KPIs such as **Completeness**, or a metric.
For **Completeness**, **Compliance** and **Correctness**:

Field	Description
Active	Activate the KPI so it is included in the aggregated CMDB health report.
Weighted Averages	Specify the weight of each metric in the aggregated KPI health report. The sum of weighted averages of all metrics should be 100.

For a metric:

Field	Description
Active	Activate the metric so it is included in the aggregated health report for the respective KPI.
Create Task	If a record fails the metric test, create a task with details about the failure. You can then view the task on the CI dashboard, and configure remediation for the task.
Failure Threshold	When the threshold number of CIs that fail the health metric test is reached, health processing stops for the metric for this cycle.
Task Assignee Group	An assignment group for the task.

4. Click **Save**.

Configure a CI attribute as mandatory so it is included in the CMDB Health tests for the required metric if enabled. Required is a metric of the CMDB Health completeness KPI.

Before you begin

Role required: itil_admin

About this task

When a field is configured as mandatory, then if the required metric is enabled, the CMDB health tests check whether that field is populated or not. The CMDB dashboard displays the aggregated report of the percentage of CIs for which one or more required fields is empty.

Procedure

1. Navigate to **All > Configuration > CI Class Manager**.
2. Click **Hierarchy** to display the CI Classes list. Then select the class with the field that needs to be set as mandatory.
3. In the class navigation bar, expand **Class Info** and then select **Attributes**. In the Attributes view, click **Added**.
4. Locate the attribute that you want to set as mandatory, and then double-click its **Mandatory** value and set it to true.

The next time the form is opened, a field status indicator appears next to the field label, indicating that a value is mandatory.

Note: Mandatory fields are global. The field is marked as mandatory everywhere it appears on a form. Also, mandatory fields do not appear correctly when using Service Mapping tag-based discovery. For more information, see [Tag-based discovery in Service Mapping](#).

Define a list of CI fields as recommended, noting that it is desirable that they are populated by a data source such as Discovery. You can then configure the CMDB completeness KPI to include recommended fields in its aggregated health reports.

Before you begin

Role required: itil has read access, itil_admin (on top of itil) has full access.

About this task

Use this for fields which should not be mandatory, but that might have useful information that the CI should have. For example, a field with information that might at some point help with diagnosis. Initially, a derived class is set with the recommended fields that are defined at the parent level. You can add or remove recommended fields for a derived class, setting it with its own recommended fields, without affecting the recommended fields at the parent or sibling levels. If all recommended fields for a derived class are removed, then the derived class automatically derives the recommended fields from its parent class.

For more information about child and parent classes, see [Table extension and classes](#).

Procedure

1. Navigate to **All > Configuration > CI Class Manager**.
2. Click **Hierarchy** to display the CI Classes list. Then select the class that contains the fields that need to be set as recommended.
3. In the class navigation bar, expand **Health** and click **Completeness**. Then click **Recommended Fields**.
4. In the Recommended Fields tab, use the slushbucket to move the fields that you want to designate as recommended, from the **Available** list to the **Selected** list.
5. Click **Save**.

Create an orphan rule to determine the percentage of orphan CIs in the CMDB. This sum is then aggregated into the correctness CMDB Health KPI, and weighed into the overall CMDB health report. Orphan rules are defined per class, and only a single orphan rule can be defined per class.

Before you begin

Role required: itil has read access, itil_admin (on top of itil) has full access.

About this task

Specify the conditions that CIs must meet to be considered an orphan CI. Specify attributes that a CI must have, relationships that a CI should not

have, or both. In the relationship conditions, you can either specify that the CI has no relationships, or a set of specific relationships that the CI doesn't have.

Note: If there is a [health inclusion rule](#) for the orphan metric, then the conditions in the health inclusion rule and the conditions in the health orphan rule, shouldn't be identical.

A health orphan rule can for example, identify a CI of the cmdb_ci_computer class as an orphan CI if the CI is not set with an owner or an asset.

Procedure

1. Navigate to **All > Configuration > CI Class Manager**.
2. Click **Hierarchy** to display the CI Classes list. Select the class for which to create an orphan rule.
3. In the class navigation bar, expand **Health**. Click **Correctness** and then click **Orphan Rule**.
4. Select a rule to edit if one exists, or click **New**. Fill out the form.

Field	Description
Class	The class for which the orphan rule applies.
Attributes	Attribute conditions that a CI must satisfy to be considered an orphan CI. For example, the filter conditions in which both the Assigned to and the Owned by fields are empty, will identify the matching CIs as orphans.
Condition	And/Or operation between the Attributes conditions and the Relationship conditions.

Field	Description
Relationship	<p>The relationship conditions that a CI must fail, based on records in the CI Relationship [cmdb_rel_ci] table, in order to be considered an orphan CI.</p> <p>To specify that a CI must have no relationships, choose Any Relation and Any Class respectively.</p>

5. Click **Submit** or **Update** to save the rule.

If the staleness metric is in effect, then staleness rules are used to determine the percentage of stale CIs in the CMDB. This sum is then aggregated into the correctness KPI, and weighs into the overall CMDB health calculation.

Before you begin

Role required: itil has read access, itil_admin (on top of itil) has full access.

About this task

The Discovery setting of certain types of CIs as stale takes precedence over a CMDB Health staleness rule defined for the CI. For more information about Discovery marking CIs as stale, see [Discovery for VMware vCenter](#).

Staleness rules are defined per class. If a rule is not defined for a class, then the parent's rule applies for that class.

Procedure

1. Navigate to **All > Configuration > CI Class Manager**.
2. Click **Hierarchy** to display the CI Classes list. Select the class for which to create a staleness rule.

3. In the class navigation bar, expand **Health** and then click **Correctness**. Click **Staleness Rule**.
4. Select a staleness rule to edit or click **New**, and then fill out the Staleness Rule form.

Field	Description
Applies to	The class for which the rule applies.
Effective Duration	<p>The time period that is used for the staleness test.</p> <p>If the CI was not updated (based on Updated sys_updated_on) within the specified time period — the CI is determined to be stale.</p> <p>If you enter a value with a prefix that is valid and a suffix that is not, such as 15 x — the valid portion of the value is used ('15'). If the entire value is invalid — the value is ignored and the previous valid value is used.</p>

5. Click **Submit** or **Update** to save the rule.

Filter the CIs that are included in health calculations and that appear in CMDB health dashboards by defining health inclusion rules. Health inclusion rules can be specified per domain.

Before you begin

Role required: itil has read access, itil_admin (on top of itil) has full access.

About this task

Evaluation for the required, orphan, recommended, duplicate and staleness health metrics, will apply only to CIs that satisfy health inclusion

rules. For example, if you want scores of the duplicate metric to appear only for server and network CIs.

Note:

- Creating health inclusion rules at the level of the base cmdb_ci table can potentially filter out health results of all classes in CMDB health dashboards.
- Applying a health inclusion rule to the duplicate metric, is supported only in the global domain.
- Due to performance issues, dot-walking in health inclusion rules for the duplicate metric is not supported.

In addition to any health inclusion rules, [identification inclusion rules](#) also indirectly impact what appears in CMDB health dashboards for duplicate CIs. The dashboard itself uses the identification engine (IRE) to identify duplicate CIs and therefore identification inclusion rules are applied.

Inheritance of health inclusion rules:

- If there are no health inclusion rules specified for a child class, then rules specified on a parent class are applied to the child class.
- If health inclusion rules are specified for a child class, then those rules take precedence over rules specified on a parent class.

In the base system, there are no predefined health inclusion rules, in which case all CIs are included in the CMDB Health calculations.

Procedure

1. Navigate to **All > Configuration > CI Class Manager**.
2. Click **Hierarchy** to display the CI Classes list. Select the class for which to create a health inclusion rule.
3. In the class navigation bar, expand **Health** and then click **Health Inclusion Rules**.
4. Click an existing rule to edit or click **New** and then fill out the Health Inclusion Rules form.

Field	Description
Applies to	Class this rules applies to.
Active record condition	Criteria that CIs must meet to be included in the evaluation for the specified health metrics.
Applies to metric	Metrics that the rule applies to.

5. Click **Save** or **Update**.

CMDB Health KPIs and metrics

The overall CMDB health score consists of three Key Performance Indicators (KPIs) which are correctness, compliance, and completeness, each further consisting of sub-metrics. Each KPI and metric is associated with a scorecard that determines its contribution to the aggregated health at the overall CMDB level, class, and CI level.

You can configure which KPIs and metrics are included in the aggregated calculation, and set their weight in the aggregation. In the base system, all KPIs and all metrics are included in the aggregated health report.

Overall

An aggregation of all three KPIs (correctness, completeness and compliance), according to their overall scorecard weight settings.

Correctness

A KPI which is an aggregation of the following metrics, according to the correctness scorecard weight settings.

Orphan

Measures the percentage of orphan CIs in the CMDB. A CI can become orphan if it was unintentionally left in the CMDB when it is no longer needed. A CI is determined to be orphan if:

- The CI satisfies the criteria in an [orphan rule](#). This criteria checks for specific attributes that a CI must have, and for CIs that have no relationships or that don't have specific relationships.
- Data is missing for the CI in its respective table, or in one of its parents' table.

Staleness

Measures the percentage of stale CIs in the CMDB. A CI is stale if it was not updated within the **Effective Duration** time period that is specified in the [staleness rule](#) that applies to the class.

The base system includes a default staleness rule for the Configuration Item [cmdb_ci] class, which sets the **Effective Duration** time to 60 days. This rule applies to all extended CMDB classes, and can be overridden by class specific staleness rules defined by the user. To determine CI staleness, a staleness rule for the CI's class is used if it exists, otherwise, the default staleness rule is used.

In addition, a relationship in which a stale CI is a parent or a child, is determined to be a stale relationship.

Note: Discovery marks VMware vCenter CIs that no longer physically exist, as stale. By default, this setting takes precedence over a CMDB Health staleness rule defined for the CI. When drilling-down in the CMDB dashboard to Health Results, the **Source** for CIs determined to be stale by Discovery, is **CLOUD_DISCOVERY**. Setting the `glide.cmdb.health.src.cmdb_health_audit_only` system property to true, ensures that the CMDB dashboard displays health results generated only by CMDB Health. For more information, see [CMDB Health system properties](#) and [Discovery for VMware vCenter](#).

Duplicate

Measures the percentage of duplicate CIs in the CMDB using [identification rules](#). Only independent CIs are evaluated for duplication. In a set of duplicate CIs, the count of duplicate CIs is the total number of CIs in the set, minus one. The detailed graphs for a duplicate set of CIs display all the CIs in the set.

For more details, examples, and troubleshooting information about duplicate metric, see the [CMDB Health - Duplicate Metric - algorithm \[KB0726425\]](#) knowledge base article.

Completeness

A KPI which is an aggregation of the following metrics, according to the completeness scorecard weight settings.

Required

Measures the percentage of CIs in which fields that are defined as mandatory, are not populated. Missing fields are tagged as incomplete noting that for this CI some information is missing. Required fields are equivalent to the fields that are [specified as mandatory](#) in the system dictionary.

Recommended

Measures the percentage of CIs in which fields that are [set as recommended](#), are not populated. Out-of-box, no recommended fields are specified.

You can use the [Add Identifier Fields In Recommended Rules](#) scheduled job to set criterion attributes from active identification rules, as recommended fields. You can use the [Remove Identifier Fields In Recommended Rules](#) scheduled job to unset criterion attributes from active identification rules, as recommended fields.

Compliance

Based on the results of actual CMDB audit runs.

Audit

[Audit](#) compares actual values of specified fields, against expected values defined in template and scripted audits. Based on the Last run date of audits, CMDB Health identifies the set of the most recent complete audit run, and uses those audit results. To pass the CMDB Health audit test, a CI must be in compliance with all audits for that CI. Create a compliance-type audit, for which the results are calculated into the CMDB Health compliance KPI.

When running [scripted audits](#), the Last run date is not populated. Therefore, for the compliance KPI to include the results of a scripted audit, update the script in the audit to record the audit run time.

Relationships

Measures the health of CI relationships, consisting of the following metrics which are not-configurable:

Duplicate relationships

Relationships that have identical parent and child CIs, identical relationship type, and an identical port. Duplicate relationships are displayed per relationship type. In a set of duplicate relationships, the duplicate relationship count is the total number of duplicate relationships in the set, minus one. The detailed graphs for a duplicate set of relationships display all the relationships in the set.

Orphan relationships

A relationship that is missing either a parent CI, a child CI, or both.

Stale relationships

A relationship in which the parent CI or the child CI is a stale CI.

A single relationship can fail more than one health test. For example, a duplicate relationship can also be stale.

Also reports the following relationship-related summaries:

- Relations not compliant with suggested relations
- Relations not compliant with containment rules
- Relations not compliant with hosting rules

Related tasks

- [View CI health](#)
- [View CI relationships health](#)
- [Create CMDB remediation rule](#)

- Configure KPI and metrics aggregation preferences
- Configure CMDB Health scorecard thresholds

Related concepts

- Domain separation in CMDB Health
- CMDB Health dashboards
- View CMDB health reports
- View services health reports
- View CMDB groups health reports

Related reference

- CMDB Health process tracking
- CMDB Health system properties

CMDB Health dashboards

CMDB dashboards display CMDB health reports and let you configure the CMDB health KPIs and metrics that CIs are evaluated for.

Dashboard	Use
CMDB Dashboard Configuration > CMDB Dashboard > CMDB View	Main CMDB health dashboard: <ul style="list-style-type: none">• Overall CMDB and class level aggregated CI health. Aggregation is displayed from the metric level up to the overall CMDB level.• Aggregated health for CI relationships, and its metrics.

Dashboard	Use
CI Dashboard <CI form> > Dashboard	<ul style="list-style-type: none"> Displays the tasks that were generated for CIs that failed a health test. Drill down for each KPI to a detailed report of associated metrics, broken by class. Manage the CMDB Health Dashboard jobs. <p>Video: CMDB Health dashboard</p>
CI Class Manager Configuration > CI Class Manager	<p>Health reports at the CI level:</p> <ul style="list-style-type: none"> Pass/fail results for each metric, per CI. Displays incidents, changes, and other tasks affecting the CI, and business services affected by the CI.
	Central location to manage CI classes and to configure CMDB health settings: <ul style="list-style-type: none"> Configure scorecard thresholds of all KPIs and associated metrics. Configure weight of KPIs and associated metrics in health aggregation. Manage rules and definitions that are used for health tests, such as orphan rules, audit certificates, and recommended fields rule.

Dashboard	Use
	<ul style="list-style-type: none"> Explore the class hierarchy. Update and extend a CI class. Delete all records for a class.
<p>CMDB Health Preferences Configuration > Health Preferences</p>	<p>Central location for configuring CMDB Health settings:</p> <ul style="list-style-type: none"> Configure CMDB Health preferences. Manage the CMDB Health Dashboard jobs. Activate and configure weighted averages for KPIs and metrics. Set the maximum failure threshold for the KPIs. Configure creation of tasks for failed CIs.
<p>CMDB Service Dashboard Configuration > CMDB Dashboard > Service View</p>	<p>Main CMDB service health dashboard:</p> <ul style="list-style-type: none"> Overall service aggregated health and detailed health for CIs per service. Aggregation is displayed from the metric level up to the overall services level. Displays the tasks that were generated for CIs in a service that failed a health test. Drill down for each KPI to a detailed report of associated metrics, broken by class.

Dashboard	Use
	<ul style="list-style-type: none">Manage the CMDB Health Dashboard jobs.
CMDB Group View Dashboard Configuration > CMDB Dashboard > Group View	Main CMDB groups (whose type is Health) dashboard: <ul style="list-style-type: none">Overall CMDB health groups aggregated health and detailed health for CIs in the group. Aggregation is displayed from the CI level up to the overall group level.Drill down for each KPI to a detailed report of associated metrics, broken by class.Manage the CMDB Health Dashboard jobs.

Related tasks

- View CI health
- View CI relationships health
- Create CMDB remediation rule

Related concepts

- Domain separation in CMDB Health
- View CMDB health reports
- View services health reports
- View CMDB groups health reports

Related reference

- CMDB Health KPIs and metrics
- CMDB Health process tracking

View CMDB health reports

The CMDB dashboard serves as a central location to view aggregated health reports for your CMDB at a glance which helps you understand the CMDB health status. Also, it provides functions to address health issues, and improve CMDB health.

The CMDB dashboard requires some configuration before it can display meaningful data. Once CMDB Health is configured and the CMDB Health Dashboard Jobs are enabled, the dashboard displays data that is automatically collected and calculated on a recurring schedule. The CMDB dashboard uses the Performance Analytics framework for dashboards and employs some of the capabilities it provides. The CMDB dashboard is domain aware.

Using the CMDB dashboard requires the asset or itil role.

For information, see:

- Sharing responsive dashboards (**Sharing**), see [Share a responsive dashboard](#).

Note: Only users with the itil role can view a CMDB dashboard which has been shared.

- How score cards in CMDB Health dashboards are calculated, see [CMDB Health Dashboard Score Card Explained \[KB0829828\]](#).

Access and configuration

Access the CMDB dashboard by navigating to **Configuration > CMDB Dashboard > CMDB View**. On the CMDB dashboard:

- Click **CMDB Health Dashboard Jobs** to enable and manage the jobs that monitor and collect health data for CIs and CI relationships.
- Click the default **CMDB Dashboard - CMDB View** dashboard to list additional CMDB drill-in dashboards.

The CMDB dashboard has two viewing modes. Click **CI Health** or **Relationship Health** to toggle between them.

CMDB Health view

The CMDB Health view is the default view for the CMDB dashboard which contains:

- Scorecards detailing the overall health of CIs in your CMDB, per health KPI and metric
- Useful reports showing a breakdown of any duplicate, orphan, or stale CIs by class
- Widgets that list the top 10 incident, alert, and change generating CIs in the CMDB

All the default widgets in the CI Class view can be filtered using the CMDB class hierarchy tree. Initially, the class hierarchy filter is set to the root class, Configuration Item (All). Click **All** to select a different class, filtering all widgets on the dashboard to display data only for the selected class and its child classes.

In each scorecard widget, the horizontal bar in the center and the % number are correlated, displaying the aggregated health summary for the KPI. Health results of associated metrics are displayed underneath, each contributing according to the configuration of the metric scorecard, and its threshold.

Except for the Overall health scorecard, you can drill into any widget in the CI Class view:

- In a scorecard widget: Click the large aggregated percent number or the health bar to drill into a more detailed dashboard for that KPI.
- In a charts widget: Click a bar to display a list of all the records that the bar represents.
- In lists: Click the 'i' icon to view a list of all the tasks or alerts related to the CI.

For more details about the CMDB Health dashboard, see the [CMDB Health Dashboard Score Card Explained](#) knowledge base article.

Relationship Health view

The Relationship Health view displays various scorecards for health indicators of CI relationships in your CMDB. It contains charts detailing any duplicate, orphan or stale relationships, broken down by relationship type. You can drill down these charts for further details.

Changing the CI Class selection while in the Relationship view has no effect on the data displayed in this view.

Color codes

Both, the CI class view and the relationship view, use color codes when displaying aggregated health status. The status definitions are based on each scorecard's threshold limits that are defined in the CI Class Manager.

Color code	Definition	Default threshold setting
Green	Best	Less than or equal to 33
Orange	At risk	More than 33 and less than or equal to 67
Red	Critical	More than 67
Gray	Incomplete	N/A

Note: The ⓘ icon is a notation that the maximum failure threshold for the scorecard has been reached. The tests for the metric are halted for this cycle, and all associated aggregated summaries are 0%. Review the scorecard rules which might be ineffective, or the CMDB might be in an unstable state.

Dashboard layout configuration

The CMDB dashboard uses some of the capabilities that Performance Analytics provides for responsive dashboards. You can, for example, add or remove widgets from the layout.

For information about adding a widget (**Add Widgets**) and changing other layout settings such as adding a tab (**Create Tab**) to the dashboard, see [Edit a dashboard](#). The drop-down that appears when adding a widget, includes CMDB-related widgets that are used in CMDB dashboards and other system widgets which are typically not relevant in CMDB reports.

Domain separation

If the Domain Support — Domain Extensions Installer plugin is activated, then the CMDB dashboard is domain aware:

- The CMDB dashboard aggregates and reports health failures and scores based on user's domain visibility of CIs. If domain visibility lets a user see a CI, then the audit rule in that user's domain applies to that CI, whether the CI is in the user's domain or in a contained domain. If a CI fails health tests from different user domains, then separate failure records are created.
- Users can configure KPI and metric settings specific to the needs in their domain. So different domains can have different settings such as active/inactive, and thresholds.
- A child domain derives its immediate parent's domain health configurations if the child domain does not configure its own. A child domain can override parent's configurations by modifying them.

Related tasks

- [View CI health](#)
- [View CI relationships health](#)
- [Create CMDB remediation rule](#)

Related concepts

- [Domain separation in CMDB Health](#)
- [CMDB Health dashboards](#)
- [View services health reports](#)
- [View CMDB groups health reports](#)

Related reference

- [CMDB Health KPIs and metrics](#)
- [CMDB Health process tracking](#)

View services health reports

The CMDB service dashboard serves as a central location to view aggregated health reports for services at a glance. Also, it lets you drill into a service to perform remediation actions that address health issues, and that improve CMDB health. The CMDB service dashboard uses the Performance Analytics framework for dashboards and employs the capabilities it provides.

Requirements

- The Event Management and Service Mapping Core plugin must be activated.
- Role required: asset or itil.

Configuration

The CMDB service dashboard requires some configuration before it can display meaningful data, using the same settings as the CMDB dashboard. The CMDB service dashboard uses the settings for the Business Service, Manual Service, and Technical Service classes. For each CI that is included in a service, the rule settings of its respective class are applied. You can customize these settings in the CI Class Manager, and on the CMDB Health Preferences page. Once CMDB Health is configured and the CMDB Health Dashboard Jobs are enabled, the CMDB service dashboard displays data that is automatically collected and calculated on a recurring schedule.

Note: The CMDB service dashboard doesn't include all services from the Service [cmdb_ci_service] table. Only services from the cmdb_ci_service_auto table and its descendants (cmdb_ci_service_discovered, cmdb_ci_service_manual, cmdb_ci_query_based_service), are included.

The `glide.cmdb.services_hierarchy_limit` system property limits the number of service CIs that appear in the CMDB service dashboard. This limit applies to any child class of the Application Services [cmdb_ci_service_auto] class and is set to 10,000 by default.

Domain separation

CMDB Health is domain aware. If domain separation has been activated, then the CMDB service dashboard displays health based on data, rules, and settings from the logged-on user domain. If rules and settings are not defined for a child domain, then the parent's settings are applied, recursively.

Access

Access the CMDB service dashboard by navigating to **Configuration > CMDB Dashboard > Service View**.

Report details

The CMDB service dashboard displays aggregated health for services, and also details for individual services. For a specific service, the CMDB service dashboard displays aggregated health for all the CIs in that service, including the service CI itself. Also it provides useful reports about service classes such as the Business Service class. You can drill down those reports to display further details of duplicate, orphan, or stale CIs per service and lists of the top 10 incident, alert, and change generating CIs in the service.

All default widgets can be filtered using the CMDB service hierarchy tree. Initially, the service hierarchy filter is set to **Business Service**. Click **Business Service** to expand it and to select a different class, filtering all widgets on the dashboard to display data only for the selected class, its child classes, or services of that class.

In each scorecard widget, the horizontal bar in the center and the % number are correlated, displaying the aggregated health summary for the KPI. Health results of associated metrics are displayed underneath, each contributing according to the weight configuration of the metric scorecard, and its threshold.

With the exception of the Overall health scorecard, you can drill into any widget in the service dashboard:

- In a scorecard widget: Click the large aggregated percent number or the health bar to drill into a more detailed dashboard for that KPI.
- In a charts widget: Click a bar to display a list of all the records that the bar represents.
- In lists: Click the 'i' icon to view a list of all the tasks or alerts related to the CI.

Color codes

The CMDB service dashboard uses color codes when displaying aggregated health status. The status definitions are based on the threshold limits for each scorecard, defined in the CI Class Manager.

Color code	Definition	Default threshold setting
Green	Best	Less than or equal to 33
Orange	At risk	More than 33 and less than or equal to 67
Red	Critical	More than 67
Gray	Incomplete	N/A

The  icon is a notation that the maximum failure threshold for the scorecard has been reached. The tests for the metric are halted for this cycle, and all associated aggregated summaries display 0%.

Related tasks

- [View CI health](#)
- [View CI relationships health](#)
- [Create CMDB remediation rule](#)

Related concepts

- [Domain separation in CMDB Health](#)
- [CMDB Health dashboards](#)
- [View CMDB health reports](#)
- [View CMDB groups health reports](#)
- [View CMDB health reports](#)
- [View CMDB groups health reports](#)

Related reference

- [CMDB Health KPIs and metrics](#)
- [CMDB Health process tracking](#)

View CMDB groups health reports

The CMDB group view dashboard serves as a central location to view aggregated health reports for CMDB groups at a glance. Also, it lets you drill into a CMDB group to perform remediation actions that address health issues, and that improve CMDB health. The CMDB group view dashboard uses the Performance Analytics framework for dashboards and employs the capabilities it provides.

Configuration

The CMDB group view dashboard requires some configuration before it can display meaningful data, using the same settings as the CMDB dashboard. For each CI that is included in a CMDB group, the rule settings of its respective class are applied. You can customize these settings in the CI Class Manager, and on the CMDB Health Preferences page. Once CMDB Health is configured and the CMDB Health Dashboard Jobs are enabled, the CMDB group view dashboard displays data that is automatically collected and calculated on a recurring schedule.

CMDB Health is domain aware. If domain separation has been activated, then the CMDB group view dashboard displays health based on data,

rules, and settings from the logged-on user domain. If rules and settings are not defined for a child domain, then the parent's settings are applied, recursively.

Role required: asset or itil.

Access

Access the CMDB group view dashboard by navigating to **Configuration > CMDB Dashboard > Group View**. Then, select a CMDB group from the **CMDB Health Group List** drop-down list.

Report details

For each CMDB group, the CMDB group view dashboard displays aggregated health for all the CIs in that group. You can drill down those reports to display further details of duplicate, orphan, or stale CIs per CMDB group. In each scorecard widget, the horizontal bar in the center and the % number are correlated, displaying the aggregated health summary for the KPI. Health results of associated metrics are displayed underneath, each contributing according to the weight configuration of the metric scorecard, and its threshold.

With the exception of the Overall health scorecard, you can drill into any widget in the CMDB group view dashboard:

- In a scorecard widget: Click the large aggregated percent number or the health bar to drill into a more detailed dashboard for that KPI.
- In a chart widget: Click a bar to display a list of all the records that the bar represents.
- In lists: Click the 'i' icon to view a list of all the tasks or alerts related to the CI.

Color codes

The CMDB group view dashboard uses color codes when displaying aggregated health status. The status definitions are based on the threshold limits for each scorecard, defined in the CI Class Manager.

Color code	Definition	Default threshold setting
Green	Best	Less than or equal to 33
Orange	At risk	More than 33 and less than or equal to 67
Red	Critical	More than 67
Gray	Incomplete	N/A

The  icon is a notation that the maximum failure threshold for the scorecard has been reached. The tests for the metric are halted for this cycle, and all associated aggregated summaries display 0%.

Related tasks

- [View CI health](#)
- [View CI relationships health](#)
- [Create CMDB remediation rule](#)
- [Configure CMDB groups scorecard thresholds](#)

Related concepts

- [Domain separation in CMDB Health](#)
- [CMDB Health dashboards](#)
- [View CMDB health reports](#)
- [View services health reports](#)
- [View CMDB health reports](#)
- [View services health reports](#)

Related reference

- CMDB Health KPIs and metrics
- CMDB Health process tracking

View CI health

The CI dashboard is a central location displaying health report for an individual CI, history of changes to the CI in a timeline view, and the relation formatter. The CI dashboard also displays incidents, changes, and other tasks affecting the CI, and business services affected by the CI. You can access the CI dashboard from a CI form, or from the CMDB dashboard.

Before you begin

Role required: asset or itil

About this task

The health scores are based on settings of CMDB Health KPIs and metrics. The report is calculated in real-time from data stored in health-related tables which the CMDB Dashboard jobs update on a recurring schedule. The completeness and correctness KPIs are always up to date, but for other KPIs, it is possible that updates to the CMDB are not reflected because one of the dashboard jobs hasn't run yet, as follows:

- Compliance: Depends on audit cycles and on the 'CMDB Health Dashboard - Compliance Score Calculation' job.
- Relationships: Depends on the 'CMDB Health Dashboard - Correctness Score Calculation' job.

To ensure that the latest updates to these KPIs are reflected on the CI dashboard, navigate to the respective dashboard job, and click **Execute Now**.

Procedure

1. On a CI form click **Dashboard**.

2. Or, navigate to **CMDB Dashboard > CMDB Health** and click **CMDB Dashboard - All** to display the class hierarchy. Enter a search string and then select a CI from the **Configuration Items** group. The search results are grouped by **Classes** and **Configuration Items** that match the search string.

Result

Various widgets in the report display CI's health with the following color codes:

- Green: The CI passed the health test (for example, it is not a duplicate).
- Red: The CI failed the health test (for example, it is a duplicate)
- Grey: The CI was not tested for this metric, because the threshold was not set for the CI (class) in the CI module.

The report displays the change history for the CI in a timeline format, that you can zoom in or out to select a time period for which to display details for. Use the related lists tabs **Change, Incident, Task, Business Services**, and **Alerts** to further drill into additional details.

Note:

Missing rules or other class definitions can prevent some health scores from being evaluated for a CI. The results in the CI dashboard in these situations, are described below:

Duplicate

- If no identification rules ([cmdb_identifier]) are defined for the CI's class or its ancestors: A notification to that effect appears.
- If only dependent identification rules are defined: Not applicable notification appears.

Orphan

- If the CI is excluded by health inclusion rules: Not applicable notification appears.
- If no orphan rules ([cmdb_health_orphan_rule]) are defined for the CI's class or its ancestors: A notification about missing a rule appears.

Staleness

- If the CI is excluded by health inclusion rules: Not applicable notification appears.
- If no staleness rules ([cmdb_health_staleness_rule]) are defined for the CI's class or its ancestors: A notification about missing a rule appears.

Audit

- If no audits ([cert_audit]) are defined for the CI (CI dashboard checks only desired states and scripted audits): Not applicable notification appears.
- If there are audits defined for the CI but the audits did not run: Not applicable notification appears.

- Customize the CI dashboard

You can add, remove or re-arrange content on the CI dashboard to display the CI health statistics that are important to you.

Related tasks

- [View CI relationships health](#)
- [Create CMDB remediation rule](#)
- [Enable and configure a CMDB Health Dashboard job](#)

Related concepts

- [Domain separation in CMDB Health](#)
- [CMDB Health dashboards](#)
- [View CMDB health reports](#)
- [View services health reports](#)
- [View CMDB groups health reports](#)
- [CI relations formatter](#)

Related reference

- [CMDB Health KPIs and metrics](#)
- [CMDB Health process tracking](#)

You can add, remove or re-arrange content on the CI dashboard to display the CI health statistics that are important to you.

Role required: `itil_admin`

On a CI form click **Dashboard** and customize the CI dashboard as follows:

- Drag a tile near its upper edge and drag it to a different location on the dashboard to rearrange the current layout.
- Click the X in the upper right side of a widget to hide the widget.

- Click the + sign in the upper left corner of the dashboard to add content. In the **Add content** dialog box, select the content to add and the location to place it.
- Click the gear icon in a widget tile to edit widget settings such as title and height.
- Click **Reset to Default** to revert to the base system settings.

View CI relationships health

View aggregated orphan, stale, and duplicate CI relationships in the CMDB dashboard. You can configure the relationship scorecards, but you cannot configure the underlying relationship KPI health tests.

Before you begin

The CMDB Health Dashboard - Relationship Compliance Processor dashboard job must run to generate data for these reports.

Role required: itil or asset

About this task

CMDB Health measures CI relationship health using a separate KPI and metrics.

Orphan relationship

A relationship that is missing parent, child, or relationship type.

Duplicate relationship

Relationships that have identical parent, child, and relationship type.

Stale relationship

A relationship in which one of the CIs is stale. For a stale CI — its associated relationships are also stale.

In addition, the following relationship compliance reports are available:

Relationships not compliant with all relationship rules

Relationships that do not comply with any [relationship governance](#) rule, including suggested relationships and dependent relationship rules.

Relationships not compliant with suggested relationships

[Suggested CI relationships](#) are used as rules to test if relationships comply with specified suggested relationships.

Relationships not compliant with containment rules

Containment rules are used to test if relationships comply with specified containment relationships.

Relationships not compliant with hosting rules

Hosting rules are used to test if relationships comply with specified hosting relationships.

For each of the compliance reports, testing a relationship requires a rule (suggested relationship, hosting rule, or containment rule) in which the parent and child CI classes match the parent and child CI classes in the tested relationship. If the relationship types in the rule and in the tested relationship do not match, then the relationship is not in compliance. If an applicable rule is not found, then the relationship is considered to be in compliance. Rules apply to the classes specified in the rule, and also to descendant classes. Therefore, when testing a relationship, rules that apply to ascendant parents of the CIs in the tested relationship are used. If there are multiple rules that match the parent and the child CI classes of the tested relationship, then the tested relationship needs to satisfy only one of these rules to be in compliance.

Procedure

1. Navigate to **All > Configuration > CMDB Dashboard > CMDB View**.
2. Select the **Relationship Health** tab.
3. Scroll to the bottom of the page to examine the relationship compliance reports.
Report results are grouped by relationship type, and you can drill down for further details:

- Point to a relationship type to display its label and the % of relationships that are not in compliance.
- Click on a relationship type to drill down to a detailed list of all the relations of that type that are not compliant. Click on a specific relationship to display more details such as the failure description. The **Failure Description** field lists only a single rule that the relationship did not comply with, even if there are additional rules that the relationship fails to comply with.

What to do next

For troubleshooting information, see the [How to identify and delete duplicate CMDB CI Relationship records, or ones that have orphan or missing parent/child relationships \[KB0780988\]](#) knowledge base article.

Related tasks

- [View CI health](#)
- [Create CMDB remediation rule](#)
- [Enable and configure a CMDB Health Dashboard job](#)

Related concepts

- [Domain separation in CMDB Health](#)
- [CMDB Health dashboards](#)
- [View CMDB health reports](#)
- [View services health reports](#)
- [View CMDB groups health reports](#)

Related reference

- [CMDB Health KPIs and metrics](#)
- [CMDB Health process tracking](#)

Create CMDB remediation rule

A CMDB remediation rule is associated with a task that was created for a failed CMDB health test. A CMDB remediation rule is applied automatically or manually to execute a remediation workflow that can, for example, delete stale CIs.

Before you begin

You must first create and publish a remediation workflow that addresses the CI issue, stored in the Workflow [wf_workflow] table. The workflow can be a regular workflow, or an Orchestration workflow, and the table in the workflow needs to match the task type in the remediation rule. Do not configure the workflow with any filter conditions by setting **If condition matches** to None, so that the filters of the CMDB remediation rule will apply.

For more information about Flow Designer (previously Workflow), see [Flow Designer](#).

Role required: itil_admin (on top of itil)

Procedure

1. Navigate to **All > Configuration > CMDB Remediations**.
2. Fill in the form.

Field	Description
Name	Remediation name.
Task type	Type of CMDB health-related tasks to apply the remediation to.
Task filter	Filters tasks to apply remediation to. Also applies dot-walking on CI fields so that remediation is applied to tasks associated with matching CIs.

Field	Description
Execution	<ul style="list-style-type: none">• Manual: Remediation is applied manually.• Automatic: The workflow is applied once, upon the creation of a task that matches the Task type and Task filter.
Active	Allowing the workflow to run.
Workflow	The CMDB remediation workflow (regular or Orchestration) that will execute automatically or manually, depending on the Execution setting. You can click the Lookup using list icon, and then click New to create a new workflow.

3. Click **Submit**.

Result

If **Execution** is set to Automatic, then the business rule Run remediations for CMDBHealth task applies the remediation workflow to CIs that match the Task filter. If **Execution** is set to Manual, then you can manually apply the remediation workflow defined in the rule.

- [Apply CMDB remediation](#)

Manually initiate a workflow to remediate a CI that failed a CMDB health test. For example, you can remediate CIs that are orphan or stale.

Related tasks

- [View CI health](#)
- [View CI relationships health](#)

Related concepts

- [Domain separation in CMDB Health](#)
- [CMDB Health dashboards](#)
- [View CMDB health reports](#)
- [View services health reports](#)
- [View CMDB groups health reports](#)

Related reference

- [CMDB Health KPIs and metrics](#)
- [CMDB Health process tracking](#)

Manually initiate a workflow to remediate a CI that failed a CMDB health test. For example, you can remediate CIs that are orphan or stale.

Before you begin

Role required: `itil_admin`

To manually apply a CMDB remediation, a CMDB remediation rule must exist, in which **Execution** is set to **Manual**.

About this task

Except for the duplicate and audit health metrics, you can choose to create tasks for health test failures for a metric.

To remediate failures of the duplicate metric, use [de-duplication tasks](#).

For all metrics except for audit, each CI that failed a metric test is associated with a single task. Because a CI can fail multiple audits, a single CI can be associated with multiple audit tasks. The first of those tasks is in the **Task** field, and any additional tasks are in the **Additional Tasks** field. To remediate failures of the audit metric, refer to the audit tasks for the audits that the CI failed.

Procedure

1. Navigate to **All > Configuration > CMDB Dashboard**, and then click **CMDB View**, **Service View**, or **Group View**.
2. Click on one of the bars in a bar chart on the page. Or, click on a metric tile that is associated with the remediation that you want to apply, and then in a detailed report click on a bar in a bar chart. For example, to remediate an orphan CI, click the **Completeness** tile.
3. In the **Task** column in the **CMDB Health Results** list, select the task that is associated with the CI that you want to remediate. Point to the information icon () for a result record to display the CMDB Health Results dialog box with more details about the health test.
The CMDB Health Results list contains records only for the CIs that failed a metric test.

Field	Column
CI	The CI associated with the test results.
Class Name	The CI's class.
Description	Details about the reasons for the CI failing the metric test.
Last Evaluated On	Time that the CI was evaluated for the metric, and which resulted in failure.
Metric	The CMDB Health metric associated with this test result.
Source	Source of the health test failure: <ul style="list-style-type: none">• CMDB Health Audit: Corresponds to the dashboard• Cloud discovery

Field	Column
Task	The task associated with the health test failure. For the audit metric, if there are multiple failures, then only the first task is listed.
Additional Tasks	If there are multiple tasks related to the audit metric, contains all tasks other than the first which is in the Task field.
Active	Used internally in combination with the To Delete field to determine the correct results set that this failure belongs to.
To Delete	Used internally in combination with the Active field to determine the correct results set that this failure belongs to.

4. On the task form, click **Remediate**.
5. In the **Run Remediations** dialog box, select the remediation rule that you want to apply.
The list of remediation rules is based on the type of health metric (such as orphan, stale), and on the filter defined in the rule.
6. Click **Execute**.

Related tasks

- [Create CMDB remediation rule](#)

Components installed with CMDB Health

Several types of components are installed with CMDB Health (included in the com.snc.cmdb plugin), such as tables, properties, and scheduled jobs.

Note: The Application Files table lists the components that are installed with this application. For instructions on how to access this table, see [Find components installed with an application](#).

Properties installed

Property	Description
glide.cmdb.services_hierarchy_limit	Maximum number of service CIs that can appear in the CMDB service dashboard. This limit applies to any child class of the Application Services [cmdb_ci_service_auto] class. <ul style="list-style-type: none">• Type: integer• Default: 10,000• Range: 0-100,000• Location: Add to System Properties [sys_properties] table.• Learn more: View services health reports

Scheduled jobs installed

Scheduled job	Description
CMDB Health Dashboard - Completeness Score Calculation	Script for calculating the completeness KPI of CMDB health.
CMDB Health Dashboard - Compliance Score Calculation	Script for calculating the compliance KPI of CMDB health.
CMDB Health Dashboard - Correctness Score Calculation	Script for calculating the correctness KPI of CMDB health.

Scheduled job	Description
CMDB Health Dashboard - Relationship Score Calculation	Script for calculating the CI relationships KPI of CMDB health.
CMDB Health Dashboard - Relationship Compliance Processor	Script for calculating compliance of relationships with suggested relationships, and with hosting and containment rules.
Add Identifier Fields In Recommended Rules	Sets all criterion attributes from all active identifier entries from all active identification rules, as recommended fields. These added recommended fields are then checked by the CMDB Health Dashboard - Completeness Score Calculation scheduled job when evaluating the recommended health metric.
Remove Identifier Fields In Recommended Rules	Identifies any recommended field that is a criterion attribute in any active identifier entry in any active identification rule. Then removes the recommended setting for that field.

Tables installed

Table	Description
CMDB Health Metric [cmdb_health_metric]	Details such as if a KPI or metric is enabled, maximum failure threshold, and other settings for all CMDB Health KPIs and metrics.
CMDB Health Result [cmdb_health_result]	Results from the most recent CMDB Health processing cycle.

Table	Description
CMDB Health Scorecard [cmdb_health_scorecard]	Current and historic health scores. Status of historic score records is 'Historic', and of latest score records is 'Complete'.
CMDB Health Orphan Rule [cmdb_health_orphan_rule]	Rules for calculating orphan records per class.
CMDB Recommended Fields [cmdb_recommended_fields]	Recommended fields per class.
CMDB Health Metric Status [cmdb_health_metric_status]	<p>Internal table that tracks the status of each KPI and metric that is being processed. Includes status, processing time, and processing start date.</p> <p>State for a KPI or metric changes from 'In Progress' to either of:</p> <ul style="list-style-type: none"> • Complete • MaxFailures • Daily Processing Time Out <p>Processing of a timed out KPI or metric continues on the following day.</p>
CMDB Health Processor Status [cmdb_health_processor_status]	<p>Internal table that tracks the processing progress of each KPI and metric. Contains a list of tables that are processed for each KPI and metric, and processing status. Classes are processed sequentially, changing status from Draft -> In Progress -> Complete.</p>

Table	Description
CMDB Relationship All Rules Health Results [cmdb_health_result_rel_all]	Stores results about relationship health, to be used by the All Relationships report.

CMDB Health process tracking

Use the following information to track and resolve issues with the CMDB Health processes.

Logging

By default, only error messages are logged to the syslog table, with the source name CmdbHealth. To enable logging of 'info' and 'warning' messages (which are typically logged at the start and end of each processing cycle), update the system property glide.cmdb.logger.use_syslog.CMDBHealth. For information about using this property, see [CMDB Health system properties](#).

Processing status

If scheduled jobs are enabled, but data is not displaying on the CMDB dashboard, you can check the processing status in the CMDB Health Metric Status [cmdb_health_metric_status] table. Depending on the status of the inactive metric, decide how to proceed.

Initially, the state of all metrics is 'In Progress'.

Possible final states of a metric:

Complete

All classes are processed and the number of failures is under the maximum failures threshold.

Max Failures

The number of failures for this metric reached the maximum failures threshold. Processing has been aborted and will start over in the next run.

Daily Time Out Pause

The processor reached the processing time limit. Processing is paused and will resume in the next run.

At the end of a processing cycle, the final state of a KPI depends on the final state of its associated metrics. Possible final state of a KPI:

Complete

All associated metrics are in Complete state and score calculation is complete.

Incomplete

Score is not calculated because one of the associated metrics reached its maximum failure thresholds.

Daily Time Out Pause

Timed out because one of the associated metrics has reached its processing time limit.

Processing time

If processing of a metric times out, you can find out which class takes too long to process. Use this information to find out if any validation rules are weak.

The progress of each metric is tracked in the CMDB Health Processor Status table [cmdb_health_processor_status]. Status for classes that have been processed for a metric is Complete, and for classes that are yet to be processed is Draft. By looking at the update time for each class, you can calculate the length of processing time for each class.

Orphan records due to broken hierarchy

Orphan rules might detect an orphan CI, which you are not able to access and delete. Or, there might be a mismatch between the list view that displays the orphan records, and the total number of records. These findings are due to records being deleted in the database from only one table in the CMDB hierarchy.

These CI records are not accessible via GlideRecord and must be deleted directly from the database. Therefore, in this case, to delete an orphan CI from the database you must contact Support to get help.

Orphan test results provide the details of where exactly the hierarchy is broken. For example, the message "This cmdb_ci_linux_server CI [91054fc24f22520053d6e1d18110c713] is missing record in cmdb_ci_computer table" means that a record of that sys_id must be deleted from the CMDB, cmdb_ci, cmdb_ci_hardware, cmdb_ci_server, and the cmdb_ci_linux_server tables (the Computer class is between the Hardware and the Server classes in the hierarchy.)

Scripted audits Skipped

An error message is logged if the results from a scripted audit are not included in the compliance KPI. The reason can be that the script in the audit was not updated to populate its **Last ran date** field. Without a **Last ran date** value, CMDB Health is unable to identify these run results as part of a recent complete audit run, and skips those results.

- CMDB Health process status: failure threshold reached

The CMDB dashboard displays the string 'failure threshold reached' when the number of CIs that are failing the metric tests, reaches the failure threshold set for the metric.

- CMDB Health process status: incomplete score

The CMDB dashboard displays the string 'incomplete score' for a metric when it fails to calculate the score for the metric.

Related tasks

- [View CI health](#)
- [View CI relationships health](#)
- [Create CMDB remediation rule](#)

Related concepts

- [Domain separation in CMDB Health](#)

- CMDB Health dashboards
- View CMDB health reports
- View services health reports
- View CMDB groups health reports

Related reference

- [CMDB Health KPIs and metrics](#)

The CMDB dashboard displays the string 'failure threshold reached' when the number of CIs that are failing the metric tests, reaches the failure threshold set for the metric.

CMDB Health stops processing for this metric in the current cycle, and therefore there is no aggregated health score for the metric. Processing will be attempted again in the next cycle. Also, status in the CMDB Health Metric Status [cmdb_health_metric_status] table is set to Max Failures for this metric.

When the health score of a metric cannot be evaluated, then the processing status of the respective KPI (for example, correctness) is set to Incomplete. The CMDB dashboard displays the string Incomplete score for the respective KPI and for the CMDB Health overall score. Also, aggregated health scores for the metric are not available for any class in the CMDB hierarchy.

Review and adjust the following definitions as needed:

- Review and refine the rules defined for the metric which has reached max failures. If a rule associated with the metric is too generic, resulting in large number of failures, attempt to refine it as follows:

•

Completeness:

Review the mandatory fields for the associated classes and remove those that aren't critical for the health score. Also, review the recommended fields that are causing failures and remove those that aren't critical for the health score.

For more information see [Set a CI attribute to be mandatory](#) and [Set a CI field to be recommended](#).

-

Compliance:

Review the audit failures from the audit jobs and adjust the audits to reduce the number of failures. For more information, see [CMDB Health KPIs and metrics](#).

-

Correctness:

- Orphan: Review the orphan rules and adjust them to remove excessive orphan failures. For more information, see [Create or edit a CMDB Health orphan rule](#).
- Staleness: Review the effective duration for the classes that are causing failures and adjust the duration to reduce the number of failures. For more information, see [Create or edit a CMDB Health staleness rule](#).
- Duplicate: Review the de-duplication tasks and remediate the tasks to remove duplicate CIs and to avoid creating failure records. For more information, see [Remediate a de-duplication task](#).
- **Increase the failure threshold** for the metric that is failing and check if processing for this metric completes successfully in the next cycle. Increasing the failure threshold beyond 500K might reduce overall performance.

The CMDB dashboard displays the string 'incomplete score' for a metric when it fails to calculate the score for the metric.

'Incomplete score' is displayed when:

- The number of CIs that are failing the tests of one of its sub-metric, reaches the failure threshold set for the metric. In this situation, the processing status for the respective parent metric (for example, correctness) is set to 'incomplete' in the CMDB Health Metric Status [cmdb_health_metric_status] table. Processing for the failing metric in the current cycle stops, and therefore there are no aggregated health

scores for the sub-metric, the parent parent metric, or the overall CMDB Health.

To remediate, you need to resolve the underlying cause of CIs failing the sub-metric tests. See [CMDB Health process status: failure threshold reached](#) for more information about resolving the failures of the sub-metric.

- An error is encountered while processing the sub-metric.

To remediate, examine the system logs to determine the cause of the error. After fixing the cause of the problem, restart processing by manually executing the respective parent metric dashboard job.

CI Class Manager

Use the CI Class Manager to centrally view, create, or edit basic class definitions, and class settings for identification, reconciliation, and CMDB Health. To access the CI Class Manager, navigate to **All > Configuration > CI Class Manager**.

Note: CI Class Manager doesn't support working with non-CMDB tables. For more details, see [IRE support for non-CMDB tables](#).

Class Basics	CMDB Health	IRE
• Create a class	• Create a compliance audit	• Identification: <ul style="list-style-type: none">• CI identification rules
• View class basic information	• Create a certification filter	• Identification inclusion rules
• Class columns	• Create a certification template	• Reconciliation: <ul style="list-style-type: none">• CI reconciliation rules
• Add a suggested relationship	• CMDB Health orphan rule	• Data refresh rules
• Display all CIs for a class	• CMDB Health staleness rules	• Dependencies:
• Dependency Views map icons for a class		
• Delete CIs		

- | | | |
|--|---|---|
| <ul style="list-style-type: none">• Update the list of classes in the Principal Class filter | <ul style="list-style-type: none">• Set a CI field to be recommended• Set a CI attribute to be mandatory• CMDB health scorecard thresholds• Health inclusion rules | Dependent relationship rules (hosting and containment) |
|--|---|---|
-

Service Mapping

- Entry point types for Service Mapping
 - CI types for Service Mapping and Discovery
-

CMDB CI Lifecycle Management (legacy)

From the time of its creation to the time that it is no longer needed, a CMDB CI would typically transition through several operational states while undergoing various operations. CI Lifecycle Management provides the mechanism to define states and actions for a CI and lets you apply appropriate actions based on a CI's state to tailor the management of CI lifecycle to business needs.

The CMDB Data Manager is now a more comprehensive and integrated solution for managing CI life cycle operations such as deletion and archival, in bulk. For information about the CMDB Data Manager, see [CMDB Data Manager](#).

Terms associated with CI Lifecycle Management:

Operational states

A set of states that a CI can be at such as 'Operational' or 'Repair in Progress'. A CI can be associated with only a single operational state at any given time. The choices for operational states are based on

the operational_status field in the [cmdb_ci] table. There are several operational states that are defined in the base system such as 'Retired' and 'Repair in Progress'. You can modify this list to reflect operational states that are relevant in your business.

Note: By default, Service Mapping is configured to ignore all host CIs for which the value of Operational status [operational_status] is not **1** (Operational) or the value of status [install_status] is **100** (absent). For additional information about this behavior, see [Preparing customized ServiceNow deployments to work with Service Mapping](#) [KB0647574] in the HI Knowledge Base.

CI Lifecycle Management allows multiple operators and automations to simultaneously set different operational states of a CI. Since a CI cannot be associated with multiple operational states, it is important to configure each operational state with a priority. These priorities are then used in such situation to determine which of the operational states is the cumulative operational state.

CI actions

A set of actions that can be applied to a CI during its lifetime. You can define CI actions that are relevant in your business.

Compatible CI Actions

CI Lifecycle Management allows a CI to have multiple active CI actions simultaneously, however they must be specifically defined as compatible. By default, there are no two actions for a CI that are compatible with each other. You can change this behavior by specifying pairs of actions that are compatible and thus allowed to be applied simultaneously to a CI. For example, you can specify that the 'Patching' and the 'Provisioning' CI actions are compatible making it possible to apply both simultaneously to a CI.

Not Allowed CI Actions

By default, any CI action can be applied to any CI. You can restrict this behavior by defining a rule that an action is not allowed for a CI when it is in a specific operational state. For example, you can define a Not Allowed CI Action in which it is not allowed to apply the 'Provisioning' action to a Linux Server that is in a 'Non-Operational' state.

Not Allowed Operational Transitions

By default, transitions are allowed from any operational state to another. You can restrict this behavior by defining a rule that for a specified CI, a transition from a certain operational state to another operational state is not allowed. For example, you can define that for a Linux Server it is not allowed to transition from 'Repair in progress' to 'Non-Operational'.

Requestor

A requestor can be a workflow or a non-workflow operator that is trying to set operational states and apply CI actions. Each requestor has an associated requestor ID that is a GUID and that can be an active workflow context or a non-workflow registered operator ID.

Lease time

A time period that each requestor (especially non-workflow operators) can provide, during which a specified CI action is allowed to be active for a specified CI.

CMDB CI Lifecycle Management provides a set of APIs to manage CI operational states and CI actions. And the UI where you define a set of rules to restrict certain operational state transitions and to restrict actions based on operational states. It also provides a mechanism to audit CI operational state and CI actions during the entire CI lifecycle.

Providers such as automation, workflows, or Change Management can use CI Lifecycle Management as a mechanism to manage CI operational states and apply CI actions. By default, the behavior of CI Lifecycle Management has no restrictions on some operations, and full restrictions on other operations. The CI Lifecycle Management UI lets you modify this default behavior by specifying Not Allowed CI Actions, Compatible CI Actions, and Not Allowed Operational Transitions that restricts some operations and enables for others.

With CI Lifecycle Management you can:

- Manage CI operational states and CI actions throughout the entire CI lifecycle.
- Manage CI operational state transitions.
- Restrict certain operational state transitions.

- Associate certain actions for certain CI types that are in specific operational state.
- Restrict IT Service Management applications based on CI operational state.
- Audit CI operational states and CI actions during the entire CI lifecycle.

Lifecycle management APIs

CI Lifecycle Management provides a set of APIs to manage CI operational state and CI actions during the entire CI lifecycle. All restrictions and allowances specified by rules in the UI are enforced when state management APIs run, and if an API attempts to perform a restricted operation, the operation is blocked and an error is logged.

Registering requestors

When using the lifecycle management APIs to apply CI actions, requestors are required to be registered and to obtain a requestor ID which is unique within the lifecycle management tables. To register and to obtain a requestor ID, non-workflow users should call the registerOperator API. Workflow users can use the active Workflow context as the requestor ID, and they do not need to explicitly call registerOperator.

After completing the CI lifecycle operations, the requestor should call the unregisterOperator API to unregister. All the state management records associated with that specific requestor ID are then marked as inactive or they are removed by the CI Lifecycle Management — Restore Internal State Management Tables scheduled job.

Integration with Incident Management and Problem Management

A base instance includes the pre-defined CI action CreateTask used for creating a task for a CI. New instances have a pre-defined Not Allowed CI Action, specifying that the 'CreateTask' action is not allowed for any CI with a **Retired** operational state. This restriction is integrated with Incident Management and with Problem Management to prevent the creation of incident or problem tasks for retired CIs. The 'CreateTask' CI action is used as a reference qualifier to the Configuration Item field of the Incident/Problem tables. In a new incident or problem, CIs in which Operational Status is 'Retired' — are filtered out from the Configuration Item list on

the form. For more information about reference qualifiers, see [Reference qualifiers](#).

Integration with Asset Management

In a base system, a CI's Operational Status field and the Status/Hardware Status (if its hardware) fields are kept synchronized if one of the two fields' values is **Retired**. When Operational Status of a CI is set to **Retired**, then the Status/Hardware Status field is automatically set to **Retired**. In the opposite direction, when the Status/Hardware Status field of a CI is set to **Retired**, Operational Status is automatically set to **Retired** too.

- When an Operational Status field changes from **Retired** to another status, the CI's Status/Hardware Status field is set to **Installed**.
-

When a CI's Status/Hardware Status field changes from **Retired** to another status, the Operational Status field is automatically set to **Non-Operational**.

The change of state from 'Retired' to another state is seldom, and by default, the state is changed to 'Non-Operational'. However, this might not be the intended state for the record. Therefore, it important that administrators review and manage the state appropriately in this case.

Whenever CI's Status/Hardware Status changes, it is synchronized to the CI's corresponding Asset State field, and vice versa — keeping the CI's Operational Status and the CI's corresponding Asset State synchronized.

For more information about mapping Asset State and Substate fields to a CI's Status/Hardware Status (if its hardware) field, see [Map asset state and CI hardware status](#). And for more information about retiring assets, see [Retire assets](#).

- [Get started with CI Lifecycle Management](#)

Follow these high level steps to get started and to track activities of the CI Lifecycle Management module of the CMDB application.

- [Lifecycle management APIs](#)

CI Lifecycle Management provides a set of state management APIs for manipulating CI operational states, and applying CI actions.

- **Components installed by CI Lifecycle Management**

Several types of components are installed by CI Lifecycle Management (included in the com.snc.cmdb plugin), including tables, scheduled jobs, and properties.

- **Activate the CI Lifecycle Management scheduled job**

When starting to use the CI Lifecycle Management module, ensure to activate the CI Lifecycle Management - Restore Internal State Management Tables scheduled job which is disabled by default. This scheduled job continuously checks and maintains the data integrity of all internal CI Lifecycle Management tables.

- **Define a CI action**

Define a CI Lifecycle Management CI action that can be later applied to CIs.

- **Define compatible CI actions**

Allow a CMDB CI Lifecycle Management operation in which two specified CI actions can be applied simultaneously to a CI.

- **Define a not-allowed CI action**

Define a restriction for CI Lifecycle Management in which a specified action is not allowed for a CI that is in a specified operational state.

- **Set priority for an operational state**

CI Lifecycle Management allows multiple operators or automations to simultaneously set different operational states for a CI. A CI can have only a single operational state, so in this case, the cumulative operational state of the CI is set to the one with the highest priority. It is recommended that you specify a priority for each operational state that you define so that a cumulative state can be correctly calculated.

- **Define a non-allowed operational transition**

Define a restriction for CI Lifecycle Management in which a specified CI cannot transition from one operational state to another.

Get started with CI Lifecycle Management

Follow these high level steps to get started and to track activities of the CI Lifecycle Management module of the CMDB application.

Before you begin

Role required: none

Procedure

1. Activate the base system [CI Lifecycle Management - Restore Internal State Management Tables](#) scheduled job that continuously checks and maintains data integrity of all internal CI Lifecycle Management tables.
2. [Define CI actions](#).
Navigate to **All > Configuration > CI Lifecycle Management > CMDB CI Actions** to display currently active/inactive CI actions in the CMDB.
3. [Define compatible CI actions](#) rules.
4. [Define not-allowed CI actions](#) rules.
5. [Define not-allowed operational state transitions](#) rules.
6. Define new operational states by modifying the operational_status field in the [cmdb_ci] table in the system dictionary.
Navigate to **All > Configuration > CI Lifecycle Management > View Internal Operational States** to display available operational states set by each requestor.
7. [Set priority for operational states](#).
8. Call APIs to apply CI actions.
Navigate to **All > Configuration > CI Lifecycle Management > CMDB CI Actions** to display which actions were submitted and thier active/inactive state in the CMDB.
9. Navigate to **All > Configuration > CI Lifecycle Management > View CI State Registered Users** to display currently registered operators that were registered via the registerOperator API.

10. Review Renew Lease tasks and extend leases as needed: Navigate to **All > Configuration > CI Lifecycle Management > Renew Lease Tasks**. These tasks are created automatically by the CI Lifecycle Management - Restore Internal State Management Tables scheduled job for CI action records in which the lease for a valid requester has expired. The Requestor should use the lifecycle management API ExtendCIActionLease to extend the lease. Otherwise, if the lease remains expired for a specified grace period, the CI Lifecycle Management - Restore Internal State Management Tables scheduled job marks the respective CI action record as 'inactive'. The grace period for expired lease time is configurable by the system property glide.cmdb.stategmt.max_lease_expired_days.
11. Navigate to **All > Configuration > CI Lifecycle Management > State Management Logs** to display logs of CI Lifecycle Management operations.

Lifecycle management APIs

CI Lifecycle Management provides a set of state management APIs for manipulating CI operational states, and applying CI actions.

State management APIs adhere to restrictions and allowances specified by Not Allowed CI Actions, Compatible CI Actions, and Not Allowed Operational Transitions. If an API attempts to perform a restricted operation, the operation is blocked, an error is logged, and a task is automatically created if appropriate.

Lifecycle management APIs can set operational states and CI actions to CMDB groups by utilizing lifecycle management bulk APIs.

Registration APIs

- `registerOperator()` - Method to register operator with state management for non-workflow user.
- `unregisterOperator(String requestorId)` - Method to unregister operator for non-workflow users.
- `isValidRequestor(String requestorId)` - Method to determine if the specified requestor is a valid active workflow user or a registered user.
- `isLeaseExpired(String requestorId, String ciSysId, String ciActionName)` - Method to check if registered user lease expired.

- `extendCIActionLease(String requestorId, String ciSysId, String ciActionName, String leaseTime)` - Method to extend CI Action Lease time, for registered users. If previous lease already expired, extend lease from now.

Operational State APIs

- `setBulkCIOperationalState(String requestorId, String sysIdList, String opsLabel, String opsStateListOld)` - Method to set Operational State for an array of CIs.
- `getOperationalState(String ciSysId)` - Method to get CI Operational State.

CI Actions APIs

- `addBulkCIAction(String requestorId, String sysIdList, String ciActionName, String ciActionListOld, String leaseTime)` - Method to add CI Action for an array of CIs.
- `removeBulkCIAction(String requestorId, String sysIdList, String ciActionName)` - Method to remove a CI Action for a list of CIs.
- `getCIActions(String ciSysId)` - Method to get CI Actions.

Not Allowed Action Based on Operational State API

`isNotAllowedAction (String ciType, String opsLabel, String actionPerformed)`
- Method to check if a specific CI action is not allowed for specific Operational State on a CI Type.

Not Allowed Operational State Transition API

`isNotAllowedOpsTransition(String ciType, String opsLabel, String transitionOpsLabel)` - Method to check if specific operational state transition is not allowed on a CI Type.

Compatible Action API

`isCompatibleCIAction(String actionPerformed, String otherActionName)`-
Method to check if two specific actions are compatible with each other.

Example: Using state management APIs

```
// 1. Register Operator with State Mgmt
var output = SNC.StateManagementScriptableApi.registerOperator();
var jsonUntil = new JSON();
var result = jsonUntil.decode(output);
var requestorId = result.requestorId;

// Get list of sys_ids to update
var sys_ids;

// 2. Set list of sys_ids's Operational State to 'Repair in Progress'
output = SNC.StateManagementScriptableApi.setBulkCIOperationalState(requestorId, sys_ids, 'Repair in Progress');
gs.print(output);

// 3. Set list of sys_ids's CI Action State to 'Patching'
output = SNC.StateManagementScriptableApi.addBulkCIAction(requestorId, sys_ids, 'Patching');
gs.print(output);
```

Components installed by CI Lifecycle Management

Several types of components are installed by CI Lifecycle Management (included in the com.snc.cmdb plugin), including tables, scheduled jobs, and properties.

Note: The Application Files table lists the components that are installed with this application. For instructions on how to access this table, see [Find components installed with an application](#).

Scheduled jobs installed

Scheduled job	Description
CI Lifecycle Management - Restore Internal State Management Tables	Continuously checks and maintains the data integrity of all

Scheduled job	Description
	internal CI Lifecycle Management tables.
Update life cycle from legacy	Updates CIs Life cycle stage and Life cycle status fields when legacy status fields change.
Update legacy from CSDM"	Updates CIs legacy status fields when life cycle changes.

Tables installed

Table	Description
CI State Registered Users [statemgmt_register_users]	All currently active registered users that were created via the registerOperator API. You cannot manually add new records to this table.
CI Actions [statemgmt_ci_actions]	A set of CI actions that can be applied to a CI during its lifetime.
CMDB CI Actions [statemgmt_cmdb_actions]	Active/inactive CI actions set by a specific requestor for a specific CI. You cannot manually add new records to this table.
Compatible CI Actions [statemgmt_compat_actions]	Set of rules that define pairs of CI actions that are compatible for a CI and can be applied simultaneously.
Not Allowed CI Actions [statemgmt_not_allow_actions]	Set of rules that define specific actions that are not allowed for a CI when it's in a specific operational state.

Table	Description
Internal Operational States [statemgmt_ops_state]	Internal operational states set by a specific active requestor for a specific CI. You cannot manually add new records to this table.
Renew Lease Task [statemgmt_renew_lease_task]	Set of tasks that were automatically created to renew the lease of CI actions whose lease has expired. You cannot manually add new records to this table.
Operational State Priorities [statemgmt_ops_state_pri]	Priorities of operational states which determine precedence when multiple operational states are set for same CIs by different requestors.
Not Allowed Operational Transitions [statemgmt_not_allow_ops]	Set of rules that define specific operational state transitions that are not allowed.

Properties installed

Property	Description
glide.cmdb.statemgmt.max_lease_expired_days	<p>Maximum number of days that lease expiration can be set with for CI Actions.</p> <ul style="list-style-type: none"> Type: integer Default value: 15 Location: System Property [sys_properties] table.

Property	Description
glide.cmdb.statemgmt.max_bulk_count	<p>Maximum number of CIs that CI Lifecycle Management can process in a bulk update operation.</p> <ul style="list-style-type: none">• Type: integer• Default value: 1000• Location: System Property [sys_properties] table.

Activate the CI Lifecycle Management scheduled job

When starting to use the CI Lifecycle Management module, ensure to activate the CI Lifecycle Management - Restore Internal State Management Tables scheduled job which is disabled by default. This scheduled job continuously checks and maintains the data integrity of all internal CI Lifecycle Management tables.

Before you begin

Role required: none

About this task

When CI Lifecycle Management operations do not complete properly, for example due to a failure of the requestor or a requestor whose lease has expired, the integrity of tables related to CI Lifecycle Management might be compromised. The CI Lifecycle Management - Restore Internal State Management Tables scheduled job scans tables related to CI Lifecycle Management, and does the following:

- De-activates or removes all internal lifecycle management records with invalid requestors, and closes any corresponding Renew Lease Tasks if present.
- Detects records associated with a valid requestor whose lease has expired, and automatically creates a Renew Lease Task to notify the

user and to provide details for extending the lease. If the requestor takes no action and the lease remains expired for a specified grace period (default 15 days), automatically de-activates the corresponding CI action record, and closes any corresponding Renew Lease Task if present.

Procedure

1. Navigate to **System Definition**, and click **Scheduled Jobs**.
2. Search for the CI Lifecycle Management - Restore Internal State Management Tables job.
3. In the respective **Active** column, double-click the value false, and select true.
4. Click the **Save** icon.

Define a CI action

Define a CI Lifecycle Management CI action that can be later applied to CIs.

Before you begin

Role required: none

About this task

You can view a list of all the actions that are currently applied to CIs by navigating to **Configuration** and clicking **CMDB CI Actions**.

Procedure

1. Navigate to **All > Configuration > CI Lifecycle Management > CI Actions**.
2. On the **CI Actions** page, click **New**. Fill in **Name** and **Description**, and then click **Submit**.

Define compatible CI actions

Allow a CMDB CI Lifecycle Management operation in which two specified CI actions can be applied simultaneously to a CI.

Before you begin

Role required: none

About this task

By default, it is not allowed to apply more than a single action to a CI. You can change that behavior by defining pairs of CI actions as compatible and therefore these actions can be applied simultaneously to a CI. For example you can specify that Provisioning and Patching are compatible CI actions, which lets you apply both to a CI at the same time.

Procedure

1. Navigate to **All > Configuration > Compatible CI Actions**.
2. On the **Compatible CI Actions** page click **New** and fill out the form.

Compatible CI Actions

Field	Description
Action	First action in the compatibility actions pair.
Compatible Action	Second action in the compatibility actions pair.

Result

An API can successfully apply the two specified actions simultaneously to a CI.

Define a not-allowed CI action

Define a restriction for CI Lifecycle Management in which a specified action is not allowed for a CI that is in a specified operational state.

Before you begin

Role required: none

About this task

By default, there are no restrictions in the CMDB CI Lifecycle Management on applying CI actions. You can restrict this behavior by not allowing a specified action to be applied to a CI when it is in a specified operational state. For example, you can define a restriction in which the provisioning action cannot be applied to a Linux Server that is in a non-operational state.

Procedure

1. Navigate to **All > Configuration > CI Lifecycle Management > Not Allowed CI Actions**.
2. Click **New** on the **Not Allowed CI Actions** page, and fill out the form.

Field	Description
Not Allowed Action	The action that is being restricted.
CI Type	The CI type for which the restriction applies to. To apply a rule to all CIs, select Configuration Item.
Operational State	The operational state that the CI must be at in order to apply the restriction.

3. Click **Submit**.

Result

If an API attempts to apply the specified action to the specified CIs, while it is in the specified operational state, the operation fails and an error is logged.

Set priority for an operational state

CI Lifecycle Management allows multiple operators or automations to simultaneously set different operational states for a CI. A CI can have only a single operational state, so in this case, the cumulative operational state of the CI is set to the one with the highest priority. It is recommended that you specify a priority for each operational state that you define so that a cumulative state can be correctly calculated.

Before you begin

Role required: none

About this task

Procedure

1. Navigate to **All > Configuration > CI Lifecycle Management > Operational State Priority**.
2. On the **Operational State Priority** page, click the operational state for which you want to set or update priority.
3. Enter a **Priority** and click **Update**.
Smaller numbers represent higher priority.

Define a non-allowed operational transition

Define a restriction for CI Lifecycle Management in which a specified CI cannot transition from one operational state to another.

Before you begin

Role required: none

About this task

By default, CI Lifecycle Management has no restrictions for transitioning CIs from one operational state to another. You can restrict this behavior by defining transitions that are not allowed for a specified CI. For example, you can define a restriction on transitioning a Linux server from non-operational state to repair in progress state.

Procedure

1. Navigate to **All > Configuration > CI Lifecycle Management > Not Allowed Operational Transitions**.
2. On the **Not Allowed Operational Transitions** page, click **New** and fill out the form.

Field	Description
CI Type	The CI type for which the restriction applies.
Not Allowed Transition	The CI state into which transitioning is restricted.
Operational State	The operational state that the CI must be in for the restriction to apply.

Result

If an API attempts to transition a CI that is in the specified operational state to a state that is not allowed, the operation fails and an error is logged.

View CMDB benchmarks

CMDB calculates several CMDB Health benchmarks which then display in the Benchmarks dashboard. These benchmarks are based on various CMDB Health metrics, displaying monthly averages, trends, comparisons to industry averages of your ServiceNow peers, and global benchmarks.

Before you begin

Role required: none

The CMDB Health Dashboard jobs must be enabled and health data must be collected. Also, navigate to **Benchmarks > Setup** and ensure that the CMDB KPIs are enabled under **IT Operations Management**.

About this task

CMDB provides the following benchmarks:

- % of non-compliant Cls
- % of duplicate Cls
- % of stale Cls

For an instance, each of these benchmarks is the calculated monthly average for the corresponding CMDB Health metric. Calculating a monthly average requires that there is a metric result value for each day of the month. Therefore, each day on which the respective Health Dashboard job did not run, is assumed with the aggregated result from the run that is most recent to that day. The monthly average is then calculated based on the sum of all the daily aggregated results for the metric in that month, divided by the number of days of the month. For a CMDB Health Dashboard job that ran multiple times in a single day, only the results of the last run in that day are used for the monthly average calculation.

Note: The frequency of CMDB Health Dashboard job executions depend on whether the job is enabled, its schedule and on manual runs.

Global averages are based on the sum of monthly averages of all peer instances, divided by the number of instances (aside from instances for which the monthly average is 0).

Procedure

1. Navigate to **Benchmarks > Dashboard**.
2. Click the  icon and select **IT Operations Management**.

3. Click **ALL** or **CMDB**.
4. Click on a CMDB benchmark to drill down to trend data, and other benchmark details.

Related concepts

- [CMDB Health](#)

CMDB APIs (CMDB SDK)

Use CMDB APIs to create, update, and read operations on the CMDB. Domain separation is supported in CMDB APIs.

CMDB APIs (CMDB SDK)

Use the following CMDB APIs to create, update, and read operations on the CMDB:

- [CMDBGroupAPI - Scoped](#)
- [CMDBTransformUtil - Global](#)
- [CMDBUtil - Global](#)
- [IdentificationEngineScriptableApi](#)
- [IdentificationEngine - Scoped](#)

Domain separation in CMDB APIs

Domain separation enables you to separate data, processes, and administrative tasks into logical groupings called domains. You can control several aspects of this separation, including which users can see and access data.

CMDB APIs are used for accessing the CMDB from a script. CMDB stores the CI and relation information; CMDB is domain separated.

CMDB APIs support the following operations:

- Create a new CI:

This operation goes through the Identification and Reconciliation Engine which supports domain separation when creating a CI. The domain of the caller is used for this operation.

- Update an existing CI:

This operation goes through the Identification and Reconciliation Engine which supports domain separation when creating a CI. The domain of the caller is used for this operation.

- Create/Delete relations:

The cmdb_rel_ci table is not domain separated.

- Query CMDB CI/Query CMDB table:

Results are filtered by the domain(s) visible to the caller.

- Query CMDB metadata table:

Metadata information is not domain separated.

Setting up domain separation for CMDB APIs

If domain separation is enabled for CMDB, then it is also available for CMDB APIs.

Data separation

Data is stored and domain separated in CMDB. There is no additional work needed from the CMDB API perspective.

Configuring a domain-separated environment

The configuration is done at the CMDB level.

If a domain column is present for base system application tables

See the [Domain separation in CMDB Health](#) topic.

Tenant domains and application data

There is no application-specific data to manage with CMDB.

Related concepts

- [Domain separation and Configuration Management Database \(CMDB\)](#)

Quick start tests for Configuration Management Database (CMDB)

Validate that Configuration Management Database (CMDB) still works after you make any configuration change such as apply an upgrade or develop an application. Copy and customize these quick start tests to pass when using your instance-specific data.

Configuration Management Database (CMDB) quick start tests require activating the Configuration Management (CMDB) plugin (com.snc.cmdb) and the CMDB - ATF Tests plugin (com.snc.cmdb.atf).

CMDB BSM: Dependency Views test suite

Test suite to check functionality of Dependency Views APIs.

Test	Description	Release version
CMDB BSM: Dependency Views	Test functionality of Dependency Views APIs. These APIs retrieve Dependency Views map and associated map items such as context menu items, for a given CI sys_id and using itil user role.	New York

CMDB HEALTH: CI Health Dashboard test suite

Test suite to check whether CMDB CI Health Dashboard is functional at a basic level.

Test	Description	Release version
CMDB HEALTH: Health Job Status	Checks whether any CMDB Health dashboard jobs, which were started 30 or more days ago, are still in progress.	New York
CMDB HEALTH: CMDB Health Completeness/Recommended	<p>Checks whether the CI dashboard is functional for the recommended metric (included in the CMDB Health completeness KPI).</p> <p>This test validates:</p> <ul style="list-style-type: none">• Creation of a health inclusion rule for the recommend metric.• Creation of a recommended field that satisfies the health inclusion rule.• Validate that the health inclusion rule is correctly applied to a test record with missing data in the recommended field.	New York

CMDB IRE: Identification Reconciliation Engine test suite

Test suite to check Identification and Reconciliation Engine (IRE) functionality.

Test	Description	Release version
CMDB IRE: IRE Validation	Validate CI identifiers and reconciliation definitions and check indexes for CI identifiers.	Madrid
CMDB IRE: Reconciliation Rule	<p>Check operations on a reconciliation rule, in CI Class Manager, using itil and itil_admin roles. Operations include create, edit, and delete a reconciliation rule.</p> <p>Also, check for active and not active setting, and derived rules.</p>	Paris
CMDB IRE: Identification Rule	<p>Check operations on an identification rule, in CI Class Manager, using itil and itil_admin roles. Operations include create, edit, and delete an identification rule.</p> <p>Also, check for active and not active setting, and derived rules.</p>	Paris

CMDB QB: Query Builder test suite

Test suite to verify CMDB Query Builder functions such as create query, read query, and execute query using two related user roles - cmdb_query_builder and cmdb_query_builder_read.

Test	Description	Release version
CMDB QB: Query Builder - cmdb_query_builder Role	Verify that cmdb_query_builder user role can save queries, and access and run all saved queries, in CMDB Query Builder.	New York
CMDB QB: Query Builder - cmdb_query_builder_read Role	Verify that cmdb_query_builder_read user role can access and run all saved queries, and cannot save any query, in CMDB Query Builder.	New York

CMDB REL: Relationship Editor and Formatter test suite

Test suite to verify functionality of Relationship Editor and Relationship Formatter.

Test	Description	Release version
CMDB REL EDITOR:Relationship Editor	Check addition of relations to a CI and deletion of relations from a CI using itil user role.	New York
CMDB REL FORMATTER:Relationship Formatter	Check accuracy of CI information, relationship types, relationships, associated records	New York

Test	Description	Release version
	such as change tickets, and settings such as CMDB views (relationship filters), displayed for a specific CI in relationship formatter using itil user role.	

CMDB SDK: SDK REST API test suite

Test suite to verify functionality of CMDB SDK Rest APIs.

Test	Description	Release version
CMDB SDK: Query CMDB Metadata	Test querying CMDB metadata.	New York
CMDB SDK: Create a relation for a CI using REST APIs	Test creation of a relationship for a CI in the CMDB using the CMDB REST APIs.	New York
CMDB SDK: Delete a relation for a CI using REST APIs	Test deletion of a relationship for a CI using CMDB REST APIs.	New York
CMDB SDK: Create a CI using REST API	Test creation of a CI using CMDB REST APIs.	New York
CMDB SDK: Query CMDB using REST APIs	Test querying the CMDB using CMDB REST APIs.	New York
CMDB SDK: Update a CI using REST APIs	Test updating of a CI using CMDB REST APIs.	New York
CMDB SDK: Query for a CI using REST APIs	Test querying a CI using CMDB REST APIs.	New York

Useful related lists in CI forms

By default, the forms that display manageable configuration items (CI) - computers, printers, network gear, uninterruptible power supplies (UPS), and power distribution units (PDU) - provide a number of related lists for the form.

The following related lists are common to all forms for manageable CIs.

- Network Adapters - Displays all the NICs installed on a CI.
- CI IPs - Displays all the IP addresses on this CI:
 - Computers (workstations, laptops using various macOS and Windows operating systems)
 - Windows servers
 - Linux servers
 - AIX servers
 - Solaris servers
 - Devices discovered through SNMP.
- DNS Names for CIs - Displays all the DNS names on a CI.

The IP version information appears in all IP address related lists and forms.

Note: Since all paths here click into the **IP Address to DNS Names** list that associates an IP address with a DNS name, this part of the common flow was not added to the tree structure.

Discovery source

A table called Source [sys_object_source] stores information identifying the source of a discovery (by ServiceNow Discovery or another product), the ID of that source, and the date/time of the last scan. To view this information, [configure a CI form](#) and add the **Sources** related list.

This table is populated automatically when the [Discovery plugin](#) is enabled.

Enterprise CMDB

The Enterprise Configuration Management Database (ECMDB) is targeted toward businesses that want to monitor, manage, measure, track, alert on change, and generally understand business systems that consist of a large number of components, business, and support personnel.

For example, a bond trading service may have multiple application, and web servers, several databases, Linux, UNIX, and Windows servers. There will be security products, network storage, disaster recovery procedures and hardware, etc. that are necessary for the service to operate properly.

The ECMDB makes it easy to either manually enter the relationships or have them populated automatically by discovery tools. In addition to the hardware, software, network, database, and storage areas, it is beneficial to know which individuals or groups are responsible for the service from both a business perspective as well as an IT perspective. Who are the line of business users and managers? Who starts and stops the application or its components? Who monitors the log files? Who is in charge of backup and restore, business continuity, and disaster recovery?

Enterprise CMDB is available with the Configuration Management (CMDB Enterprise Edition) (com.snc.cmdb.enterprise) plugin, which is active in the base system.

- [CMDB Relationships](#)

The ECMDB lets you easily track all relationships by relationship type.

- [Enterprise Configuration Management Database \(ECMDB\) action icons](#)

Any of the following icons may appear in the ECMDB lists of related items.

- [Business service tables](#)

In the CMDB, child tables of the Service table [cmdb_ci_service] store information about services, including application services.

CMDB Relationships

The ECMDB lets you easily track all relationships by relationship type.

The Enterprise CMDB extends the capabilities of the ServiceNow platform CMDB in the following areas.

Extended configuration item types

- Clusters
- Database Instances (Oracle, MySQL, MSFT SQL Server)
- File Systems (Direct and network attached)
- Linux Servers
- Solaris Servers
- AIX Servers

Extended relationships

Accurate description of relationships between items, and between items and people or groups, is important to understand the fabric of a business service. ECMDB provides many relationship types out of the box, but it is easy to extend the number of relationship types. Example relationship types include the following.

- Connects to
- Depends on / Provides Service to
- Powered by / Powers
- Protected by / Protects
- Disaster Recovery Provided by / Provides Disaster Recovery for

For descriptions of some key relationships, see [CI relationships in the CMDB](#).

Visualization

The system can show relationships as a hierarchy using a standard treeview, flattened, or graphically, all in a simple web interface.

Auditing

Auditing of changes to configuration items is turned on by default.

Federation

Federation of third party discovery and configuration data is supported through [Service Graph Connectors](#).

Configuration item modeling (product models)

Model driven configuration management allows the definition of CI models up front that can be associated to product maintenance lifecycles, cost centers, and support organizations, as well as provides a means for capacity and inventory planning. By defining models for CIs (which have a many to one relationship to the model), you can dynamically group actual discovered or imported CIs into logical, operational, and financial models. This facilitates an organized approach to managing your assets (CIs) in their respective domains.

Related reference

- [Enterprise Configuration Management Database \(ECMDB\) action icons](#)
- [Business service tables](#)

Enterprise Configuration Management Database (ECMDB) action icons

Any of the following icons may appear in the ECMDB lists of related items.



For currently active incidents against this configuration item



For currently active problems against the configuration item



For currently active changes against the configuration item that are not covered in the past, current, pending changes. For example, a request to update the operating system on a server that is currently in progress may display this icon.



For changes that were recently completed against the configuration item. changes with an "Actual_end_date" in the past.



For changes that are planned soon against the configuration item. changes with an "Actual start date" in the future.



For currently active changes against the configuration item that have an "Actual start date"



For outages that were recently completed against the configuration item. outages with an "end" date in the past.



For outages that are planned soon against the configuration item. outages with a "begin" date in the future.



For currently active outages against the configuration item that have a "begin" date in the past and no "end" date



This will only show up in the Tree view and indicates that a configuration item that is downstream has at least one of the above issues against it.

The system looks 5 calendar days in the past and 7 calendar days in the future when looking at recent outages and changes.

Related reference

- [CMDB Relationships](#)
- [Business service tables](#)

Business service tables

In the CMDB, child tables of the Service table [cmdb_ci_service] store information about services, including application services.

An application service is work or goods that are supported by an IT infrastructure. For example, delivering email service to an employee can require services such as email servers, web servers, and the work to configure the user's account. An application service management map graphically displays the configuration items (CI) that support an application service and the relationships between the CIs.

For information on some other CMDB tables, see [CMDB schema model](#).

Key fields in the Service table

Field	Description
Business criticality	<p>The importance of this service to the business. This field can be used to determine disaster recovery strategies for this service. Default options are:</p> <ul style="list-style-type: none">• 1 - most critical• 2 - somewhat critical• 3 - less critical• 4 - not critical

Field	Description
SLA	A reference to the Agreement [sla] table.
Service classification	<p>Designates the type of the service.</p> <ul style="list-style-type: none"> • Application Service (For more information, see Application services) • Technical Service (For more information, see Application services and Populate an application service using the Dynamic CI Group method) • Service Offering (For more information, see Service Portfolio Management service offerings) • Shared Service (For more information, see IT shared services) • Billable Service (To represents a service that is billed, or that is cost managed)
Used for	<p>Designates how this service is used. Default options are:</p> <ul style="list-style-type: none"> • Production • Staging • QA • Test • Development • Demonstration

Field	Description
	<ul style="list-style-type: none">• Training• Disaster Recovery
Users supported	The users that this service supports. A reference to the Group [sys_users_group] table.
Version	Use this field for your own versioning processes.

The Service Configuration Item Association table

The Service Configuration Item Association table [svc_ci_assoc] binds an application service and a CI to track which CIs are part of each application service.

The Service Configuration Item Association table

Field	Description
Configuration Item Id	A reference to the Configuration Item [cmdb_ci] table.
Service Id	A reference to the Service [cmdb_ci_service] table.

Related reference

- CMDB Relationships
- Enterprise Configuration Management Database (ECMDB) action icons

Table form views

When you view a table definition form, you can open the context menu, and select a form view in which to display the table. The default view for

a table is the Default view. For any class that is an extension of the CMDB table, you can select the CI Definition view which provides additional access to related tables and information.

The CI Definition form view is a centralized location from which you can configure and view a table. In addition to the information that the default view displays, the CI Definition form view provides the following controls.

Control	Description
Icon tab	View and create new NG-BSM icons for CI types
CI Identifier tab	View and create new CI identifiers
Reconciliation Definitions tab	View and create new data source definitions
Inclusion related link	Links to the Metadata Editor

To access these additional controls on the CI Definition form view, you need to first create a new table that is derived from the CMDB table, and then view it using the CI Definition form.

Related tasks

- [Create a CI class](#)

Platform Analytics Solution for Configuration Management (CMDB)

Platform Analytics Solutions contain preconfigured dashboards. These dashboards contain actionable data visualizations that help you improve your business processes and practices.

Use the Platform Analytics widgets on the dashboard to visualize data over time, analyze your business processes, and identify areas of improvement. With Platform Analytics Solutions, you can get value from

Performance Analytics for your application with minimal setup. You can always create your own objects as well.

Important: Set up and test Platform Analytics Solutions on a non-production instance before enabling them in production.

To enable the solutions for Configuration Management (CMDB), an admin can navigate to **Performance Analytics > Guided Setup**. Click **Get Started** then scroll to the section for Configuration Management (CMDB). The guided setup takes you through the entire setup and configuration process.

Common Service Data Model

The CSDM is the data framework that you follow when you set up ServiceNow products and applications. You adhere to the CSDM guidelines when you define configuration items (CIs) and relationships between CIs in the CMDB. This process ensures that your data resides in the appropriate CMDB tables for maximum value from your Now Platform applications.

About the CSDM

The CSDM is the data model standard for all products that use the Configuration Management Database (CMDB).

- The CSDM guidelines ensure unified data access for Now Platform products.
- The CSDM gives you clear direct prescriptive guidelines for service modeling within the CMDB.
- CSDM terms and definitions ensure consistent and accurate service reporting.
- The CSDM data model supports multiple configuration strategies and includes guidelines for using base-system tables and relationships.
- You can use the CMDB query builder to create reports showing CMDB configuration items (CIs) and their relationships.

You will find additional information about the CSDM on the [Community Forum](#). Also, see [CSDM in a nutshell](#). [View the full list of CSDM and data foundation videos](#).

Expert guidance to assess and improve your CSDM implementation — the CSDM Assessment

The CSDM Assessment provides Impact Customers with leading practices and prescriptive guidance on the CSDM and how it supports processes on the Now Platform. To help your organization plan for and implement CSDM, the assessment includes interactions with ServiceNow CSDM experts and personalized content. See [Common Service Data Model \(CSDM\) Assessment](#).

CSDM documentation



Explore

CSDM in a nutshell

Exploring the CSDM framework

View the full list of CSDM and data foundation videos.

CSDM hub in the ServiceNow Community

Implement the framework



Implementing the CSDM framework in stages

Manage the framework



Managing the CSDM framework



Reference

CSDM reference

Applying the CSDM guidelines to
your product

Key principles that guided the design and development of the CSDM framework

The framework is intended to help you avoid errors in implementation and to ensure that your Now Platform products generate accurate, complete, and consistent reports. The following principles guided the development of the CSDM framework.

Use simplified concepts

Represent concepts in a simple, distinct manner to eliminate duplicates and confusion over data sources.

Design for reporting and analytics

A prime objective of CSDM is to support consistent analysis.

Prescribe the data relationships

Tell users in a clear direct way which relationships and references to use to link CSDM tables.

Share the data model across products

The CSDM identifies a data model that is shared across products to simplify concepts and collaboration. Collaborating with other product teams achieves the best shared design.

Use clear definitions

Use agreed-upon CSDM definitions wherever a table, reference, or attribute is used.

Share base-system tables

zBoot must provide shared base-system CSDM tables by default.

Consistent data integrations

To ensure data integrity, use prescribed technologies when integrating external data sources

Speed adoption

For each new release, provide automation and guidance for CSDM that accelerates upgrading and minimizes issues.

Enable data governance and process

The presence of data within the model provides little value without governance and effective process to manage the truth and validity of the data.

Provide practical user documentation

(The content that you're viewing now) Each product team that references CSDM objects should provide documented guidance on use and/or value of the objects.

- [Exploring the CSDM framework](#)

Learn more about the CSDM.

- [Implementing the CSDM framework in stages](#)

Following the CSDM framework ensures that you meet your primary goal of consistent accuracy in reporting and analytics so you can effectively manage your digital environment.

- [Managing the CSDM framework](#)

The CSDM is the data framework that you follow when you set up ServiceNow products and applications. You adhere to the CSDM guidelines when you define configuration items (CIs) and relationships between CIs in the CMDB. This process ensures that your data resides in the appropriate CMDB tables for maximum value from your Now Platform applications.

- [Applying the CSDM guidelines to your product](#)

The "Product view" topics describe how several ServiceNow products benefit from your use of the Common Service Data Model (CSDM) framework.

- [CSDM reference](#)

The following sections show the terms, tables, and relationships in the CSDM.

Exploring the CSDM framework

Learn more about the CSDM.

CSDM overview

The CSDM is the data framework that you follow when you set up ServiceNow products and applications. You adhere to the CSDM guidelines when you define configuration items (CIs) and relationships between CIs in the CMDB. This process ensures that your data resides in the appropriate CMDB tables for maximum value from your Now Platform applications.

[Join the ServiceNow CSDM Community to learn more.](#)

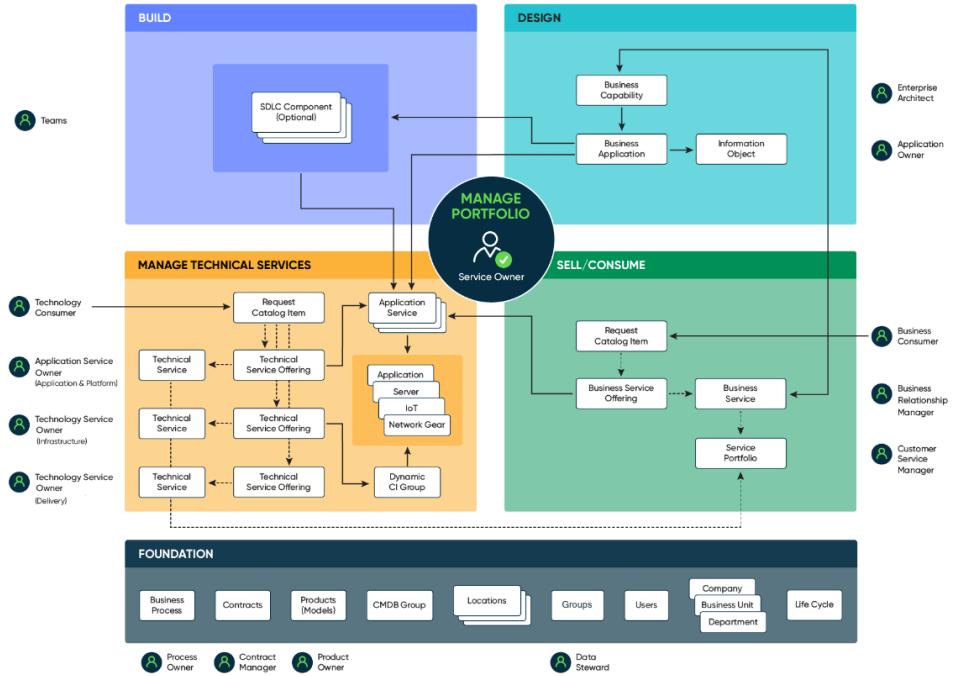
Definitions of CSDM terms

See [CSDM terms](#).

CSDM conceptual model

This illustration identifies each CSDM domain and the several roles and user types that work together to manage your ServiceNow applications and services. Each domain is associated with one or more products, services, or service types.

See [Common Service Data Model — conceptual model](#) for a full description.



CSDM benefits

Benefit	Value and associated features
Rationalization — Determine how well an enterprise application is performing	<p>Identify insights from operational activities related to services (visualize aggregated information across all your services and supporting operations).</p> <p>Application Portfolio Management (APM) and IT Business Management (ITBM)</p>
Outage reduction — Identify affected business services quickly	<p>Prioritize critical events based on service-related information.</p> <p>IT Service Management (ITSM) and Event Management</p>

Benefit	Value and associated features
Reporting — Consistent and reliable information about your services and digital products	Understand Services, their makeup and accountabilities. IT Service Management (ITSM)
Alignment — Align business processes to reduce risk and assure compliance	Understand Service Health (availability, CSAT, Performance, Vulnerabilities, IPC stats) Governance, Risk, and Compliance (GRC)

CSDM videos in the ServiceNow Community

[View the full list of CSDM and data foundation videos.](#)

Selected videos:

[Data Foundations dashboards for CSDM and CMDB](#)

[CSDM in a nutshell](#)

[CSDM 4.0 What's New](#)

[CSDM V4: Product and life cycle discussion](#)

[Application service types](#)

[Ask the Expert: ITSM Common Services Data Model \(CSDM\)](#)

[Technical service vs. Business service](#)

[Digital Portfolio Management](#)

[CSDM Technical service — Deep dive](#)

[CSDM V4 and APM](#)

[CSDM V4 and Service Builder](#)

[CSDM Example Series: Platforms](#)

CSDM Example Series: Microservices

CSDM Example Series: Shared tech and client compute services

CSDM / CMDB discussion: Extending the CMDB and support for the full life cycle

Business criticality

Creating a report on Business Critical Application Services using CMDB Query Builder

TechTalk: Build a rock solid digital foundation (CMDB) with ITOM Visibility — Deep dive

Digital Product and CSDM

End-to-end cloud app creation and management

Assignment, change, support and managed by groups in CSDM

ServiceNow Epic Council April 23 2021

How Change Management leverages the CSDM

- Common Service Data Model — conceptual model

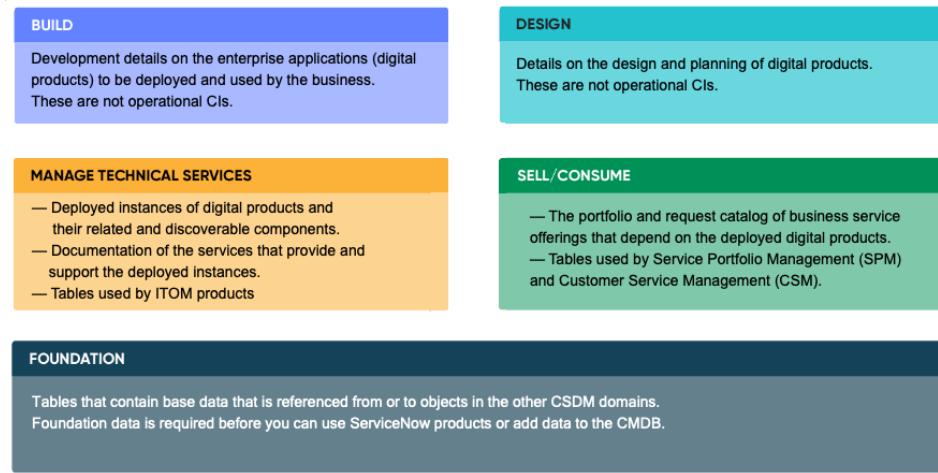
The CSDM is the data framework that you follow when you set up ServiceNow products and applications. You adhere to the CSDM guidelines when you define configuration items (CIs) and relationships between CIs in the CMDB. This process ensures that your data resides in the appropriate CMDB tables for maximum value from your Now Platform applications.

Common Service Data Model — conceptual model

The CSDM is the data framework that you follow when you set up ServiceNow products and applications. You adhere to the CSDM guidelines when you define configuration items (CIs) and relationships between CIs in the CMDB. This process ensures that your data resides in the appropriate CMDB tables for maximum value from your Now Platform applications.

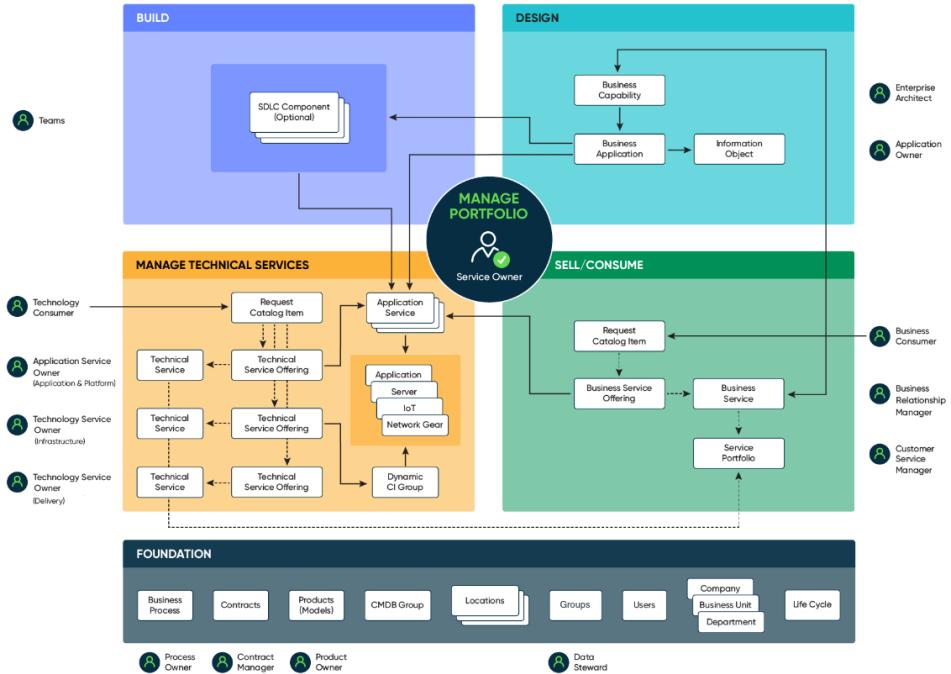
CSDM domains: Overview

This illustration of the CSDM conceptual model describes the CSDM domains at a high level.



CSDM domains: Details

Each domain is associated with one or more products, services, or service types, each of which you can extend as needed. Every box in the diagram (except Request Catalog Item) represents Cls in the CMDB. Roles and user types appear next to their area of responsibility.



- The Foundation domain involves tables that contain base data that is referenced from or to objects in the other CSDM domains. Foundation data is required before you can use ServiceNow products or add data to the CMDB. See [Foundation domain of the CSDM framework](#)
- The Design domain supports the design and planning of digital products. CIs in the Design domain aren't operational, so you can't select them for Incident Management, Problem Management, or Change Management. Enterprise architects and application owners are the typical users of tables in this domain. See [Design domain of the CSDM framework](#).
- The Build domain involves the tables that are used in the build effort (systems development life cycle — SDLC or Agile Development) of digital products like DevOps. The tables represent the logical development details of the enterprise applications to be deployed and used by the business. See [Build domain of the CSDM framework](#).
- The Manage Technical Services domain involves the tables used by IT Operations Management (ITOM) products such as Service Mapping

and Discovery. See [Manage Technical Services domain of the CSDM framework](#).

- The Sell/Consume domain involves the tables used by Service Portfolio Management (Service Portfolio Management) and Customer Service Management (CSM). See [Sell/Consume domain of the CSDM framework](#).
- The Manage Portfolio domain is a layer on top of the CSDM conceptual model that interacts with the other CSDM domains. The typical user, a service owner, might be responsible for services in more than one domain. See [Manage Portfolio domain of the CSDM framework](#).

Services and service types

A service enables you to achieve the outcomes that you want with minimal risks and without incurring costs. This definition is consistent with the base definition of “service” in ITIL v3 and IT4IT. Services typically have three components: the interaction, the offering, and the service system.

The Now Platform includes the following base-system service types that you can extend to align with the service types in your organization.

Application services ([Application Service table \[cmdb_ci_service_auto\]](#))

Application services are logical representations of a deployed application stack. Because application services are logical in nature, they should use the Logical life cycle states. Application services follow the same life cycle guidance as any other logical CI.

- An application service is an operational CI and a unique instance of an application.
- Used in Incident, Problem, and Change.
- Can be created for each region and each environment (Development, QA, and Production).
- Can be created via manual mapping, service mapping with entry point, and dynamic query.

For more information about leaf nodes and structured hierarchies, see [Design domain of the CSDM framework](#).

Business services (`cmdb_ci_service_business`)

A business service is a service type that is published to business users. A business service typically implements one or more business capabilities.

Usually, business users order business services. Business users can select the desired offering and service commitment levels via the Service Catalog. For example, procurement, shipping, and finance.

- A business service is an operational CI.
- A business service must be a one-level service and not a hierarchy of business services.
- A business service can be used for impact in Incident, Problem, and Change and for approvals for Change.
- A business service must be focused on the consumer or seller.

Technical services (`cmdb_ci_service_technical`)

Technical services are the systems associated with the admins of CIs in the Manage Technical Services domain: Application service owners, Technical service owners, and Technology service owners). Technical services are typically lower-level leaf nodes of one or more business services or application services in a structured hierarchy.

- Technical services are operational CIs.
- A technical service must be a one-level service and not a hierarchy of technical services.
- Technical services are used for impact in Incident, Problem, and Change. Also used for approvals for Change.
- Technical services must be provider-focused and include the technology provided for the business to consume or sell.

CSDM videos in the ServiceNow Community

[CSDM terms](#)

[Playlist of all CSDM and data foundation videos](#)

[CSDM in a nutshell](#)

CSDM V4: What's New

CSDM V4: Product and life cycle discussion

Application service types

Ask the Expert: ITSM Common Services Data Model (CSDM)

Technical service vs. Business service

Digital Portfolio Management

CSDM Technical service — Deep dive

CSDM V4 and APM

CSDM V4 and Service Builder

CSDM Example Series: Platforms

CSDM Example Series: Microservices

CSDM Example Series: Shared tech and client compute services

CSDM / CMDB discussion: Extending the CMDB and support for the full life cycle

Business criticality

Creating a report on Business Critical Application Services using CMDB Query Builder

TechTalk: Build a rock solid digital foundation (CMDB) with ITOM Visibility — Deep dive

Digital Product and CSDM

End-to-end cloud app creation and management

Assignment, change, support and managed by groups in CSDM

ServiceNow Epic Council April 23 2021

How Change Management leverages the CSDM

- **Foundation domain of the CSDM framework**

The Foundation domain involves tables that contain base data that is referenced from or to objects in the other CSDM domains. Foundation data is required before you can use ServiceNow products or add data to the CMDB.

- **Design domain of the CSDM framework**

The Design domain supports the design and planning of digital products. Cls in the Design domain aren't operational, so you can't select them for Incident Management, Problem Management, or Change Management. Enterprise architects and application owners are the typical users of tables in this domain.

- **Build domain of the CSDM framework**

The Build domain involves the tables that are used in the build effort (systems development life cycle — SDLC or Agile Development) of digital products like DevOps. The tables represent the logical development details of the enterprise applications to be deployed and used by the business.

- **Manage Technical Services domain of the CSDM framework**

The Manage Technical Services domain involves the tables used by IT Operations Management (ITOM) products such as Service Mapping and Discovery.

- **Sell/Consume domain of the CSDM framework**

The Sell/Consume domain represents the portfolio of business services that may sell or consume elements of the Manage Technical Services domain. The Sell/Consume domain involves the tables used by Service Portfolio Management (Service Portfolio Management) and Customer Service Management (CSM).

- **Manage Portfolio domain of the CSDM framework**

The Manage Portfolio domain is a layer on top of the CSDM conceptual model that interacts with the other CSDM domains. The typical user, a service owner, might be responsible for services in more than one domain.

The Foundation domain involves tables that contain base data that is referenced from or to objects in the other CSDM domains. Foundation data is required before you can use ServiceNow products or add data to the CMDB.

The tables in the Foundation domain aren't used in Configuration Management Database (CMDB) relationships. Instead, the tables contain critical referential data. Typical users of the domain are process owners, data stewards, product owners, and contract managers.



Business process

A business process has a well-defined start and finish. Examples of business processes in the banking industry are the customer onboarding process and the credit check process. Each business process can have levels of criticality and impact. Business processes are stored in the `cmdb_ci_business_process` table.

In a parent-child relationship, business processes can be identified by using the `parent` attribute as a reference to a parent business process.

The business process is a manually-maintained CI that can identify declared and determined criticality as well as impact to confidentiality, integrity, and availability. Business processes can be reviewed monthly, quarterly, semi-annually, or annually. In addition, the next review date can be recorded. For further information, see [Business process management](#) and [Create a business process](#).

Contracts

A contract is a binding agreement between two parties. In the Now Platform, contracts contain detailed information such as the contract number, start and end dates, active status, terms and conditions statements, documents, renewal information, and financial terms.

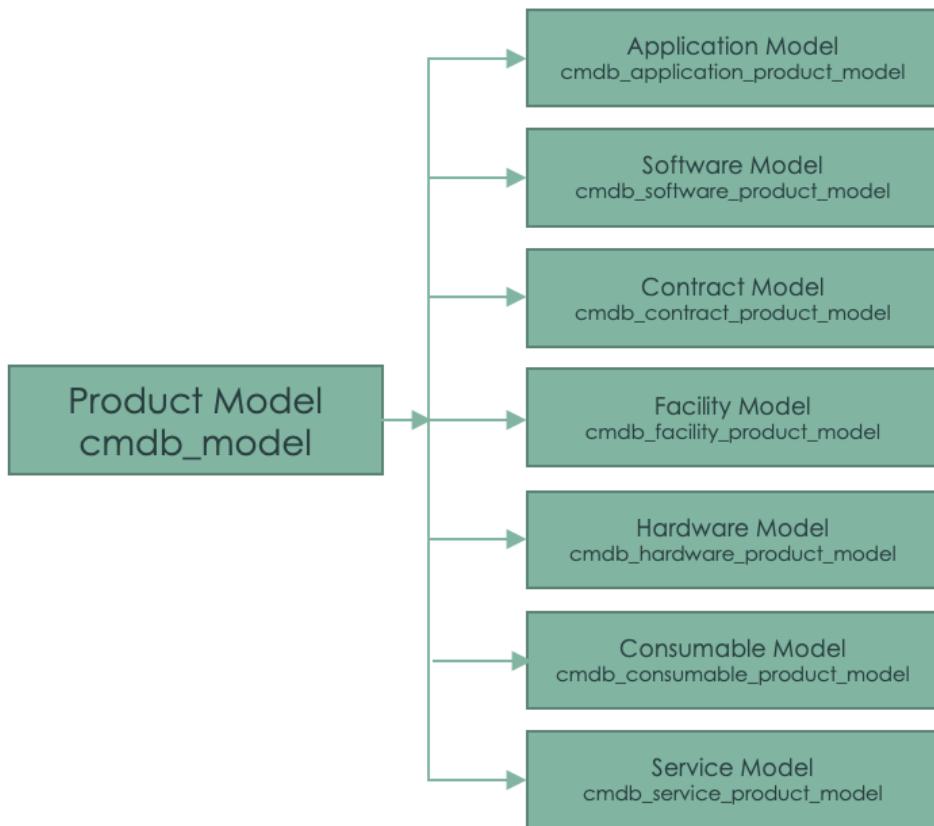
- A contract is not a CI. Contracts use contract model types from the [Product Models](#) module. Contracts are stored in the [ast_contract] table.
- Use the Contract Management application to manage and track contracts. See [Contract Management application](#).
- In the Service Level Management application, contracts group together SLAs that relate to a single vendor or customer, as well as the CIs, locations, groups, users, and child contracts that are related to the contract. For more information, see [Define a service contract](#).
- Service contracts used by Vendor Management Workspace can support hardware CIs as part of an SLA.
- In the Customer Service Management product, service contracts define the type of support that customers receive. A contract can include an account and contact or a consumer and the specific assets that are covered. A contract can also include multiple service entitlements and SLAs. See [Define a service contract in Customer Service Management](#).

Products and product models

A product model is a specific version or configuration of a product used to manage and track applications on the Now Platform. Product models identify the product owner, team, product status, compatibility with other products, reference to product catalog, and reference objects in the various stages of a product's life cycle. For more information, see [Product catalog](#).

Additionally, you can identify the products reaching end-of-life as defined by third party providers or internal product owners. You can also bundle other products as components to represent the set of products that your organization develops, sells, or uses.

Product models are extended into seven base types: application model (version agnostic), software model (version specific), contract model, facility model, hardware model, consumable model, service model. Products might be bundled to create a collection or group of products, for example a FlashBlade server (hardware model), or a 24/7 support service (service model).



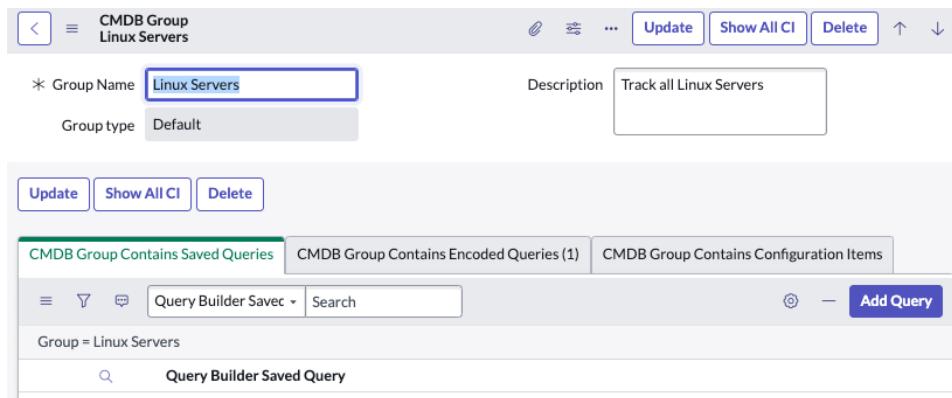
Product models are stored in the [`cmdb_model`] table or the extended tables aligned to the seven base types. The product model tables are not CIs. Configuration items can use the **Model ID** attribute to reference product models. For example, a service offering CI might reference a particular service model that other service offerings of the same type also reference.

Application, service, and software class instance CIs aren't created through Discovery, so their **Model ID [model_id]** values might not refer to product model records. To help you to migrate to a product-centric management paradigm, each instance of a logical CI should be associated with a product model. See [Auto-generate product models for logical CIs](#).

CMDB group

A CMDB group is a collection of CIs (but is not, itself, a CI). A group is based on the results of saved Query Builder queries, encoded queries, or manual entries. You can apply an action to all members of a group at one time.

You can work with a CMDB group across the Now Platform.



- For the CSDM, the Dynamic CI Group references a CMDB group to provide a list of CIs based on a common criteria.
- CMDB groups are stored in the table [cmdb_group].
- The CMDB group can potentially replace the spreadsheets that you might be using to group your CIs.

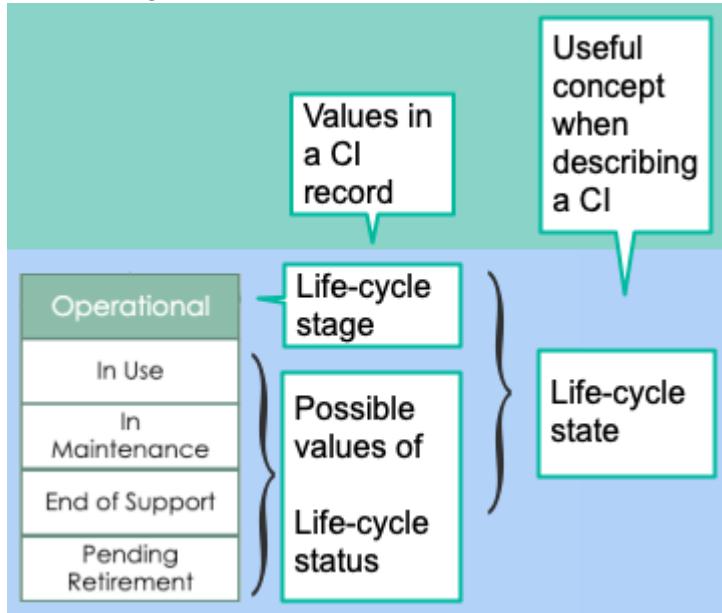
For additional information, see [CMDB groups](#).

Life-cycle states

Life-cycle states track the life cycles for products, assets, contracts, CIs, locations, and other objects. Using the standard CSDM life-cycle values consistently helps you to effectively track objects through their transitions over time. Reporting can therefore accurately reflect the actual states of CIs: usage, availability, end of support, and so on.

See the ServiceNow Community video: [CSDM V4 product and life cycle discussion](#)

You can think of the life-cycle state as the combination of two values of an asset or CI: the life-cycle stage and life-cycle status over the CI's life cycle. For example, a hardware CI in the **Operational** stage might change status over time from **In Use** to **In Maintenance** to **End of Support**. A different hardware CI might go from **In Use** to **End of Support** without ever having been in **In Maintenance** status.



When you enable the CSDM framework, you can start using the Life Cycle Stage and Life Cycle Stage Status fields to track an asset's life cycle. To use the fields, follow the procedure described in [Second activation step — Activate the CSDM plugin](#). The following life-cycle processes can use life-cycle fields:

- Product life cycle
- Hardware life cycle
- Logical life cycle
- Document life cycle
- Location life cycle

Legacy life-cycle statuses are auto-updated

The following legacy statuses are automatically mapped to the Life Cycle Stage and Life Cycle Stage Status fields when you follow the procedure described in [Second activation step — Activate the CSDM plugin](#).

Important: Legacy fields are not deleted after you implement the Life Cycle Stage and Life Cycle Stage Status fields.

- Product Model Status
- Asset State
- Asset Substate
- Contract Status
- CI Install Status
- CI Operational Status
- CI Hardware Status
- CI Hardware Substatus

Map your existing life-cycle values to CSDM life-cycle states

Use the Life Cycle Mapping module ([CSDM > Life Cycle Mapping](#)) to specify how your existing life-cycle values should be converted to CSDM life-cycle states. The mapping ensures Now Platform products "see" legacy CIs in your environment. In this example, the existing **Pending Install** value of the **Install Status** attribute for hardware CIs will always map to the **Deploy/Test** life-cycle state in the CMDB. See [Specify how to map legacy life-cycle states to CMDB states](#) and [Activate life cycle migration](#).

* Mapping for table **Hardware [cmdb_ci.hardware]**

* Priority **10**

Active **Active**

Fields

* Legacy field name	Install Status	Life cycle control	Hardware - Deploy - Test
* Legacy field value	Pending Install	Table	Hardware
Legacy subfield name		Life Cycle Stage	Deploy
Legacy subfield value		Life Cycle Stage Status	Test

Common data

Common data elements are not configuration items. Common data is shared and used throughout the Now Platform. Common data includes organizational structure (Company, Business Unit, Department), locations, groups, and users. Many Now Platform products depend on common data to provide business value.

Planning your common data is essential to the effective implementation of Now Platform products and features. Consider the following issues:

- Do you have a trusted source for the data?
- Do you have multiple data sources?
- How often does the data change?
- Do you have the depth of data that the CIs require?
- Who maintains the data?

Common data is stored in the following tables:

- Company: [core_company]

- Business unit: [business_unit]
- Department: [cmn_department]
- Location: [cmn_location]
- Groups: [sys_user_group]
- Users: [sys_user]

Location management

Data that comes from multiple sources and federated integrations is difficult to maintain. The following attributes have been added to the location (cmn_location) table to simplify management:

- **Source:** The origin of the location record.
- **Location type:** The position of the location record in the hierarchy of locations. You can use the following options to create a hierarchy of location data to suit your requirements: Region, County, State/Province, City, Site, Building/Structure, Floor, and Room.
- **Managed by group:** The group that governs or manages this location record.
- **Validation** (duplicate and primary): Flag duplicate records and manually filter locations that are not be displayed.
- **Life cycle stage** and **Life cycle stage status:** See [Life-cycle states](#).

CSDM videos in the ServiceNow Community

[CSDM terms](#)

[CSDM V4: Product and life cycle discussion](#)

[CSDM 4.0 What's New](#)

[CSDM / CMDB discussion: Extending the CMDB and support for the full life cycle](#)

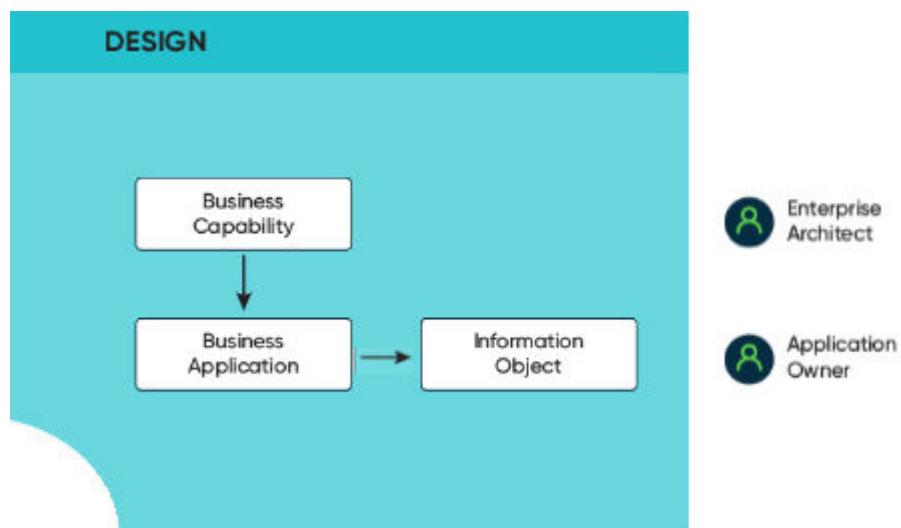
[View the full list of CSDM and data foundation videos.](#)

The Design domain supports the design and planning of digital products. CIs in the Design domain aren't operational, so you can't select them for Incident Management, Problem Management, or Change Management. Enterprise architects and application owners are the typical users of tables in this domain.

Tables used in the Design domain

The Design domain includes the tables used by Application Portfolio Management (APM). You use APM to rationalize and manage your business applications, but you're not required to use APM to benefit from the data in these tables.

- Business capability table [cmdb_ci_business_capability]
- Business application table [cmdb_ci_business_app]
- Information object table [cmdb_ci_information_object]



Relationships between CIs that support decision making

An accurate service model that includes the following relationships can serve as the foundation for strategically aligned architectural decisions. The relationships enable you to determine the risks involved in using particular business capabilities. In addition, they enable you to assess

services based on their relationships to business capabilities and business applications.

Relationship between a business capability and its supporting business applications

This relationship supports visualization and reporting.

-

A business capability is a high-level capability that supports a business model or fulfills a mission for your organization.

A business capability typically describes a specific task that achieves one or more business outcomes. Business capabilities are often listed as verbs (for example, manage financials or provide IT support services). You can use business capabilities to rationalize and prioritize the cost of business applications and business services.

- A business application represents the software and infrastructure that provides a business function (for example, the titles catalog). Business applications are not strictly required, but they are strongly recommended because they increase productivity and perform other business functions such as accounts payables, accounts receivables, and general ledger. You can use APM to add any business application for which you must track costs, usage, business value, functionality, and risks.

Relationship between a business application and the application services

The relationship connects the record of the business application that is used in planning and design with where and how it's realized operationally, represented by application services. The relationship accounts for each use of a business application in the development, test, and production environments (dev, test, and prod application service instances). Often there are multiple production deployments. For example, a large retailer uses a business application that runs a cash register in each of its 1,000 stores. There are therefore 1,000 production instances of the application service — one per store — for that one business application. See the "CSDM in a nutshell" video for additional discussion of the relationship.

Business capabilities represented in a hierarchy

You can represent business capabilities in a hierarchy of a parent business capability and one or more lower-level (child) capabilities. Child capabilities (leaf nodes) are represented by numeric values: 1.0 for the parent and 2.0 through 6.0 for the leaf nodes. If you add, update, or delete a capability at a leaf node, be sure to update the levels of all the capabilities for the leaf nodes in that hierarchy, as applicable. If a business capability hierarchy requires more than six levels, divide the structure into multiple business capabilities.

Use the Business Capability form to create, modify, and extend business capabilities.

The screenshot shows the ServiceNow interface for managing business capabilities. At the top, there's a header bar with a back arrow, a list icon, three dots, and buttons for Dashboard, Form (which is selected), Update, and Delete. The main area has a title "Business Capability" and a subtitle "Manage supplier recovery". Below this, there are fields for Name (with "Manage supplier recovery" entered), Parent (set to "Service products after sales"), Level (set to 2), Business Unit, and Department. A Description field is also present. A "Related Items" section includes a search bar for CI and buttons for Create, Import, and Export. At the bottom of this section are "Update" and "Delete" buttons. Below this is a "Related Links" section with links to "Multisource Data Preview" and "Subscribe". The bottom part of the screenshot shows a list of capabilities under the heading "Parent = Manage supplier recovery". The list has columns for Name, Description, Level, and Order. It displays the message "No records to display". There's also a "New" button and a "Related Links" section with a link to "Update Capability Level and HierarchyID".

To update capabilities, select the **Update Capability Level and HierarchyID** related link. Follow these guidelines when you update capabilities to ensure that the capability map reflects the change:

- If the parent capability is updated in the hierarchy, the levels of all its leaf node capabilities are recalculated.
- The total number of leaf node levels in a hierarchy can't exceed six.
- When adding a capability, the hierarchy level is automatically assigned based on the parent capability level.
- You can delete only leaf node-level capabilities or capabilities without leaf node levels.
- Don't create circular relationships. For instance, when creating a parent capability, a leaf node capability can't be its parent.

Adding a Business application

A business application is a manually managed CI class. You must therefore manually create required relationships to CIs (for example, with instances of the application services in use). Creating relationships also enables you to relate business applications to infrastructure CIs such as databases and web servers. If needed, you can integrate or connect two or more business applications to establish their relationship.

A business application can span the following entities:

- Environments (for example, Development, Test, or Production).
- Geographies (for example, the Americas, the Asia Pacific Japan (APJ)
- Regions (for example, Europe, the Middle East, and Africa [EMEA]).

Use either of the following methods to add a business application:

- Import the list of applications from a spreadsheet or third-party tool. To import data, define a data source and a transform map and then run or schedule an import.
- Use the Business Application form.

Business Application
New record [Business Application ReadOnly view]

* Name	Compliance Training	Status	In Production
Business process	Learning Path	Technology stack	Java
Application type	COTS	User base	1000+
Architecture type	Web Based	Platform	Oracle
Install type	Cloud	Last change applied date	2018-10-01
Business Unit	Legal		
Department	Legal		
* Description	The compliance training business application focuses on security, SEC rules, trade policies and other regulatory training for all employees.		
<input checked="" type="radio"/> Owners <input type="radio"/> Compliance			
Business owner	Fred Luddy	Last updated by	
* IT Application owner	Abel Tuter		
<input type="button" value="Submit"/>			

Information object

- An information object is a CI that displays and describes the information (or type of data) that the application receives from the database. Information objects are part of the information portfolio and are referenced by the business application.
- Information objects are mapped to the information object table [cmdb_ci_information_object].
- You can use the information object table to identify the types of data that a business application uses, including highly sensitive data such as:
 - Personally Identifiable Information (PII)
 - Payment Card Industry Data Security Standard (PCI DSS) data
 - Health Insurance Portability and Accountability Act (HIPAA) data

CSDM videos in the ServiceNow Community

[CSDM terms](#)

End-to-end cloud app creation and management

Technical service vs. Business service

CSDM V4 and Service Builder

Business criticality

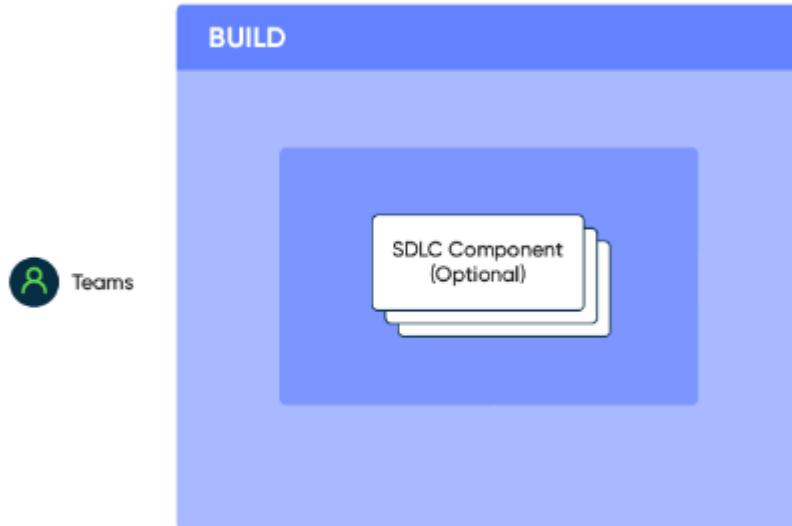
[View the full list of CSDM and data foundation videos.](#)

The Build domain involves the tables that are used in the build effort (systems development life cycle — SDLC or Agile Development) of digital products like DevOps. The tables represent the logical development details of the enterprise applications to be deployed and used by the business.

SDLC component CI records in the SDLC Component table [cmdb_ci_sdlc_component] enable the DevOps product to provide enhanced capabilities for visualizing and managing your application development pipeline.

Records in the table are not operational and are not direct targets of the ITSM Incident Management, Problem Management, or Change Management processes. You therefore are not required to configure SDLC component records.

The tables in the build domain reference the logical development details of the enterprise applications to be deployed and used by the business. A common persona in this domain is Teams. The SDLC Component table is available through the CMDB schema version 1.33.



SDLC component

An SDLC component is a CI that represents a unique development effort of code. It represents parts of a larger business application or digital product broken down into its individually developed components. In other words, the SDLC component is a software element of a larger application or technology.

Types of SDLC components:

- Application: An application service is a deployed instance of the SDLC application component. Examples include micro services and APIs. The build team typically builds application services on behalf of the Service Owner (as described in [Manage Portfolio domain of the CSDM framework](#)).
- Infrastructure: Any infrastructure CI that represents a snapshot of its configuration details is a deployed instance of the SDLC infrastructure component. Examples include database and security configurations.

CSDM videos in the ServiceNow Community

[CSDM terms](#)

[End-to-end cloud app creation and management](#)

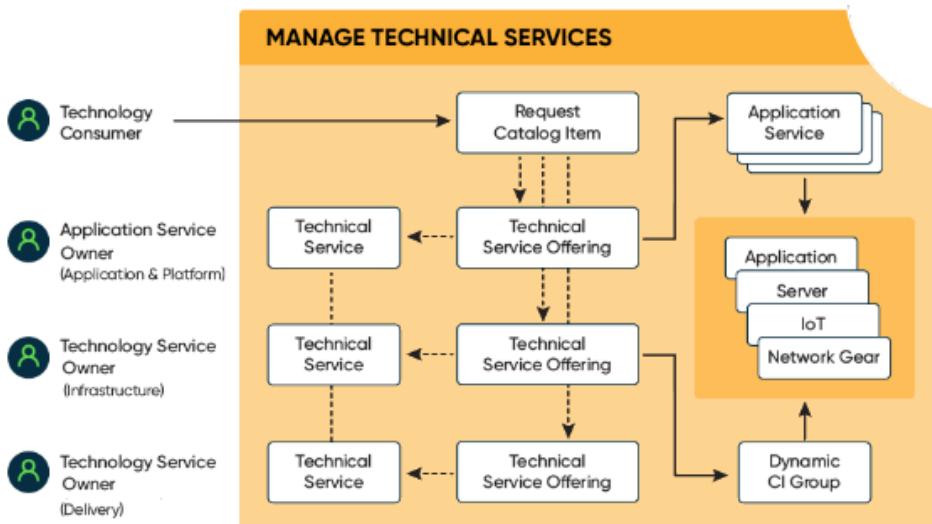
CSDM V4 and Service Builder

[View the full list of CSDM and data foundation videos.](#)

The Manage Technical Services domain involves the tables used by IT Operations Management (ITOM) products such as Service Mapping and Discovery.

The CIs in this domain are discovered items such as installed applications, servers, and network components. The Manage Technical Services domain also represents the portfolio of technical services in use. These services are operational, which means that you can select them for ITSM Incident Management, Problem Management, or Change Management.

Typical users are application service owners (for the application and platform) and technology service owners (for the infrastructure and delivery). Technology consumers can request technical service offerings through the [Request Catalog](#).



The tables in the Manage Technical Services domain represent the technology that your business sells or consumes in the provider view. While you aren't required to use Service Mapping and Discovery to populate the tables, those products accelerate the process and minimize errors. They also enable you to manage Cls and their relationships. The domain includes the following tables:

- Technical service table [cmdb_ci_service_technical]
- Technical services in Event Management use the Dynamic CI group table [cmdb_ci_query_based_service].
- Request catalog
- Technical service offering table [service_offering]
- Dynamic CI group table [cmdb_ci_query_based_service]
- Mapped Application Service table [cmdb_ci_service_discovered] (included in the base system)

Technical services

Technical services are associated with service owners and are typically layered under one or more business or application services. A technical service may have one or more technical service offerings.

Technical service users can view and manage the technologies that you provide to the business. Event Management enables you to monitor service performance. You can also use Event Management to identify health issues for related infrastructure CIs and application services.

Technical services can be managed as part of the Service Portfolio in the Sell/Consume domain (that is, a Service Portfolio hierarchy can be referenced from a Technical Service). This allows for a more complete hierarchy and management of both Technical Services and Business Services within the Service Portfolio Management workspace and related workspaces. You can make better decisions when you know how spend on technical services can improve performance and reliability of your business services.

Technical service offerings

Technology consumers can request technical service offerings (SO) through the [Request Catalog](#). The consumer can typically select the following features and options:

- Level of performance
- Location or geography
- Environment

- Pricing
- Availability
- Capability
- Support group (for incident)
- Technical approval group (for change)
- Packaging options (commitments)

Technical service offerings typically have the following components:

One or more service commitments

A service commitment defines the service delivery obligations agreed to between the consumer and the provider. Service commitments uniquely define the level of service in terms of availability, criticality, scope, pricing, and other factors. For example, an organization may offer two levels of support for an application service:

- Support for a production-level offering: Provides a high level of availability and criticality for production instances. Includes a 24/7, 5-minute response time guarantee (24 hours per day seven days per week).
- Support for a non-production-level offering: Limited availability and criticality for non-production instances. Includes a 60-minute response time guarantee between 8:00 a.m. and 5:00 p.m., Monday through Friday.

A service offering subscription that records which users have access to an offering

Technical service offerings that are mapped to the [service_offering] table are classified as "technical service" and are derived from the service. The technical service offering is based on how the parent serves a specific technical need. Every operational technical service should have at least one technical service offering.

Each CI associated through a Dynamic CI Group can be related to only one Technical Service or Technical Service Offering. Conflicts can result when one service includes multiple offerings with different SLAs, OLAs, Support Groups, and commitments.

Dynamic CI groups

A dynamic CI group is comprised of CIs that result from a CMDB Groups query. For example, you can create a dynamic CI group based on location: "all web servers in Detroit" or "all Oracle databases in Mumbai".

Note: Dynamic CI groups contain only CIs and can't contain other CI groups.

Dynamic CI group are mapped to the [cmdb_ci_query_based_service] table and are classified as either application service or technical service, as applicable. You might want to use dynamic CI groups in the following situations:

Query-based application service

You don't have Service Mapping enabled yet, but you have 12 servers and three database instances in MyAppServiceProd. You can replace your spreadsheets with a dynamic CI group as an application service.

Managed group of Infrastructure CIs

The web servers in Detroit are managed by the DetroitRockCity Technical Service Offering. Instead of manually creating relationships from Technical Service Offerings to Infrastructure CIs, use a Dynamic CI group. A single relationship from your Technical Service Offering CI (DetroitRockCity) to your dynamic CI Group (web servers in Detroit) gives you the visibility you need.

A way to manage patches for your CIs

In Change Management, you can select the dynamic CI group for the CIs you need to update and use a business rule to auto-populate the **Affected CI** field.

See [Create a Dynamic CI Group](#) for instructions.

Application services

An application service is a logical representation of a deployed system or an application stack. Using application services, you can view maps and change history for services. For example, the Event Management application can monitor service performance and identify health issues for application services.

Application services can be internal, like an organization's email system or customer-facing, like an organization's website. For example, creating financial reports through a web-based application requires a computer, web server, application server, databases, middleware, and network infrastructure. The applications and hosts are configured to offer the service of financial reporting. An application service represents an instance of such a business application or system in the development, test, or production environment.

Application services are the entry points for the Service Mapping feature. Application services underpin a business or technical service and are mapped to the CMDB Application Service table [cmdb_ci_service_auto] for common reporting.

Application services are key relationship entities for IT Service Management (ITSM), IT Operations Management (ITOM), Strategic Portfolio Management (SPM), and Customer Service Management (CSM).

Application services include relationships between business applications, business services, technical services, applications, and infrastructure CIs. You can expose an application service by using the related business or technical service offering.

The table that an application service maps to depends on the method used to create it:

Methods mapped to tables

Method used to create the application service	Mapped to table
Top Down Discovery (Service Mapping)	cmdb_ci_service_discovered
Dynamic CI Group (Query-based)	cmdb_ci_query_based_service
Tags	cmdb_ci_service_tags
Manual (using the Create an Application Service form)	cmdb_ci_service_discovered

- For more information about application services and the methods you can use to create them, see [Application services](#) and [Create an application service](#).
- You can specify required attributes for application services. See [Specifying attributes and relationships for Application Services](#) and [Modify the attributes and relationships required for application services](#).
- You can set a relationship between an application service and the components of other CSDM domains. See [Service Mapping](#).

Applications

An application is any program or module that defines behavior and performs a specific function. Applications are typically discoverable instances and provide a specific set of functions for one or more services.

- The application table and extended tables contain uniquely discovered instances of code in use on the host.
- Applications are considered infrastructure CIs.
- The instance is limited to the applications on a single host. This limitation ensures that applications are uniquely identified during discovery.
- There's a one-to-many (and not a one-to-one) relationship between the application and the application service. A single installed application, such as a database instance, may support multiple application services depending on the configuration and the use of the applications.

Note: The application table [cmdb_ci_appl] isn't an inventory or portfolio of your applications. Don't make the mistake of storing managed application details in the application table. Those details (inventory or application portfolio objects) belong in the business application table (as documented in [Design domain of the CSDM framework](#)).

Infrastructure CIs

Infrastructure CIs are managed physical and logical components. A CI can be a single module, such as a server, database, or a router or a complete system such as a web server, database, or infrastructure.

The underlying infrastructure components or Cls can be complicated. The complexity increases as data structures are layered on top of physical Cls. For that reason, you should work with a business relationship manager or enterprise architect to define your business capabilities and business applications.

CSDM videos in the ServiceNow Community

[CSDM terms](#)

[Technical service vs. Business service](#)

[CSDM Technical service — Deep dive](#)

[CSDM V4 and Service Builder](#)

[CSDM Example Series: Microservices](#)

[CSDM Example Series: Shared tech and client compute services](#)

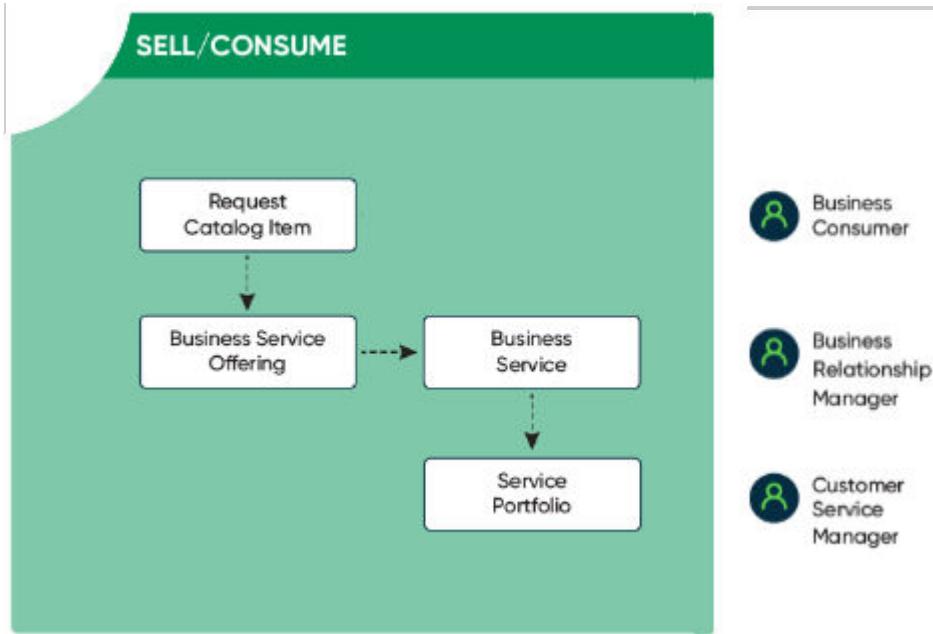
[Creating a report on Business Critical Application Services using CMDB Query Builder](#)

[Application service types](#)

[View the full list of CSDM and data foundation videos.](#)

The Sell/Consume domain represents the portfolio of business services that may sell or consume elements of the Manage Technical Services domain. The Sell/Consume domain involves the tables used by Service Portfolio Management (Service Portfolio Management) and Customer Service Management (CSM).

Typical users are the business relationship manager and the customer service manager. Business consumers can request business services through the [Request Catalog](#). You're not required to use Service Portfolio Management or CSM to use the referenced tables, however those products enable you to manage workflows and report service-related data.



The Sell/Consume domain includes the following tables:

- Business service offering table [service_offering].
- The Business Service table [cmdb_ci_service_business] extends the core Service table [cmdb_ci_service].

Note: Before the Business Service table was added, all Business Services existed in the Service table. In the future, all Business Services might migrate from core cmdb_ci_service to cmdb_ci_service_business. Until then, both tables operate identically.

- Service portfolio table [spm_service_portfolio]. The Service portfolio table is not a CMDB table.

You can select the tables in the Sell/Consume domain to use with Incident Management and Change Management.

Business service offerings

Business service offerings are the starting point for configuring Service Portfolio Management. Business service offerings inherit from Business

Services. Business service offerings consist of one or more service commitments that define the level of service in terms of availability, scope, pricing, and other factors. For example, an organization might offer two levels of desktop support:

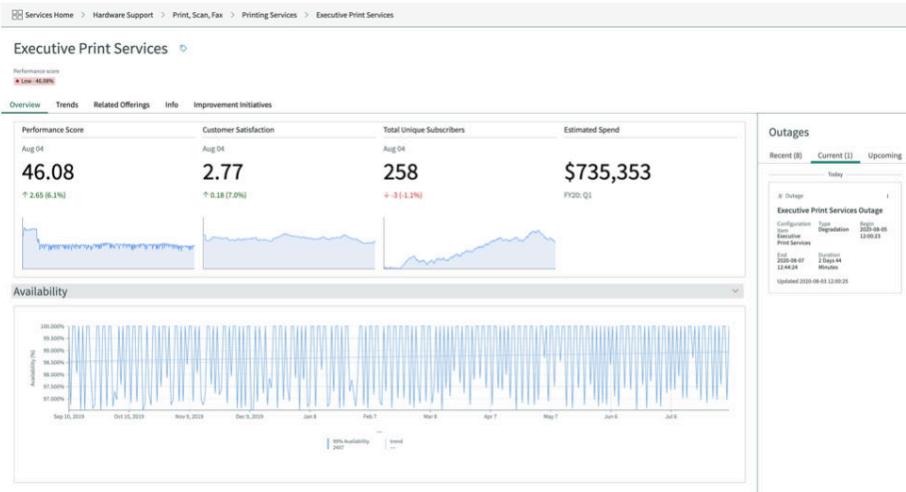
- A silver offering of upgrades and virus protection.
- A gold offering with the silver commitments plus a response time guarantee of 30 minutes between the hours of 8:00 a.m. and 5:00 p.m., Monday through Friday.

Business service offerings have the following characteristics:

- Business service offerings tailor the service by capability, availability, pricing, and packaging options. You can use the service offering to set different levels of performance and features for a particular service.
- Business service offering commitments define the agreed-upon service delivery obligations.
- Business service offering subscriptions record which users have access to an offering.
- Business service offerings are the CMDB records that identify the specific business area and the entity where the service is delivered. Some business services and service offerings depend on the application service.
- Business service offerings are derived from the service and are refined depending on how the parent serves a particular business need.

Note: You should configure at least one service offering for each operational business service or technical service.

You can view your business service offerings in the Digital Portfolio Management (DPM).



Business service offerings typically have different service-level agreements (SLAs) depending on their commitments. Without a business service offering, SLAs remain at a process level only. For example, the SLA stays at a P1 incident or a minor change, and doesn't refer to the affected service offering.

You can represent business services and business service offerings as catalog items in the service catalog to make them available for consumers.

Business services

A business service is associated with business users and is typically layered beneath one or more business capabilities. A business service can contain one or more business service offerings.

Business consumers can use the [Request Catalog](#) to order business services, business service offerings, and service commitment levels. Business services are mapped to the [cmdb_ci_service_business] table and are classified as “business services.”

Service portfolios

A service portfolio is a hierarchical collection of business services, products, projects, or applications. A portfolio can represent a strategic business objective and enables you to manage all included items as a

group (for example, life cycles). Items are organized into the following categories:

- Objective (business intent)
- Capability
- Organization (for example, enterprise resource planning [ERP] or financial management)
- Geography (location)

Request catalogs

A request catalog enables consumers to order and manage business and technical products, services, service commitment options, and offerings (for example, the Human Resources [HR] service catalog). Catalogs contain catalog items and are the starting point for consumers to access available services.

Catalog Item

A catalog item is an item or a service that a consumer can request from the catalog. A service can contain multiple catalog items (for example, the employee onboarding catalog). Catalog items are listed on the service portal and are available to the users that need them (either through subscription or job responsibility). Each catalog item is linked to one service offering.

CSDM videos in the ServiceNow Community

[CSDM terms](#)

[Technical service vs. Business service](#)

[CSDM V4 and Service Builder](#)

[CSDM Example Series: Microservices](#)

[CSDM Example Series: Shared tech and client compute services](#)

[View the full list of CSDM and data foundation videos.](#)

The Manage Portfolio domain is a layer on top of the CSDM conceptual model that interacts with the other CSDM domains. The typical user,

a service owner, might be responsible for services in more than one domain.



For example, in the Sell/Consume domain, the service owner for Human Resources (HR) might be financially responsible for the business application that provides HR services. The service owner might also need to manage the HR application instances (known as application services or systems) and might also be accountable for the impact the application has on the business.

Because of these additional responsibilities, CSDM enables service owners to oversee business applications and their deployed instances. This visibility enables service owners to perform their duties and meet their responsibilities.

CSDM videos in the ServiceNow Community

[CSDM terms](#)

[Digital Portfolio Management](#)

[CSDM V4 and APM](#)

[View the full list of CSDM and data foundation videos.](#)

Implementing the CSDM framework in stages

Following the CSDM framework ensures that you meet your primary goal of consistent accuracy in reporting and analytics so you can effectively manage your digital environment.

Improve your CSDM implementation

The CSDM Assessment provides Impact Customers with leading practices and prescriptive guidance on the CSDM and how it supports processes on the Now Platform. To help your organization plan for and implement CSDM, the assessment includes interactions with ServiceNow CSDM experts and personalized content. See [Common Service Data Model \(CSDM\) Assessment](#).

Activating CSDM

First activation step — Map existing life cycle data to CSDM standards

The CSDM enforces standard life-cycle states to ensure that assets are tracked accurately over life-cycle transitions. You migrate all life-cycle data across the platform to the CSDM standard.

Second activation step — Activate the CSDM plugin

Activate the CSDM plugin so you can begin implementing the framework.

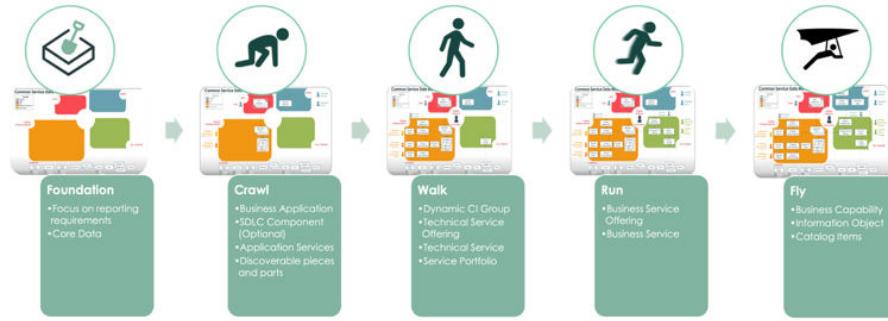
Third activation step — Migrate existing data to the CSDM framework

You complete several tasks to ensure that your existing application data migrates successfully to the required tables in the CMDB.

Implementing the CSDM framework

It's best to use a staged approach when you implement the CSDM framework. Each implementation stage involves particular information types and provides specific benefits. Because each stage builds on the preceding stage, we use an analogy to the way a person develops:

foundation, crawl, walk, run, and, eventually, fly.



Note: Business applications reference information objects in the information portfolio. You might need to implement the Information Object table [cmdb_ci_information_object] earlier than the Fly stage. Your business requirements determine the right stage for implementing the table.

CSDM implementation stages — Foundation

In the Foundation stage of implementing the CSDM framework, you prepare the referential data that enables accurate reporting to support good business decisions. Use the base-system tables when you begin implementing the CSDM to derive the highest value from your ServiceNow products and the Now Platform.

CSDM implementation stages — Crawl

In the Crawl stage, you work on base-system CMDB tables that are associated with IT Service Management (ITSM).

CSDM implementation stages — Walk

In the Walk stage, you identify and populate the network infrastructure CIs and applications that your organization's technical teams support.

CSDM implementation stages — Run

In the Run stage, you set up the relationship between a technology and the business that sells and/or consumes the technology.

CSDM implementation stages — Fly

When you reach the Fly stage, you've accomplished all or most of the process of implementing the CSDM framework. The fly stage completes the process.

Key guidelines for you to follow

- When linking CSDM tables, use only the relationships that are designed in the model.
- Collaborate on the shared data model with other product teams. Also, when you extend CSDM and related functionality, be sure to follow the provided guidance. Following the guidance and collaborating with other product teams helps you achieve the best design.
- Use agreed-upon CSDM definitions whenever you use a table, reference, or attribute.
- Use the provided CSDM base tables.
- Use the recommended technologies when you integrate external data sources. The specified process ensures data integrity and integration consistency.
- Follow the provided guidance for setting up and using Now Platform products.
- Configure the CSDM Data Foundations dashboard

Use the CSDM Data Foundations dashboard to monitor and evaluate key foundational metrics of the CSDM framework.

- First activation step — Map existing life cycle data to CSDM standards

The CSDM enforces standard life-cycle states to ensure that assets are tracked accurately over life-cycle transitions. You migrate all life-cycle data across the platform to the CSDM standard.

- Second activation step — Activate the CSDM plugin

Activate the CSDM plugin so you can begin implementing the framework.

- Third activation step — Migrate existing data to the CSDM framework

You complete several tasks to ensure that your existing application data migrates successfully to the required tables in the CMDB.

- **CSDM implementation stages — Foundation**

In the Foundation stage of implementing the CSDM framework, you prepare the referential data that enables accurate reporting to support good business decisions. Use the base-system tables when you begin implementing the CSDM to derive the highest value from your ServiceNow products and the Now Platform.

- **CSDM implementation stages — Crawl**

In the Crawl stage, you work on base-system CMDB tables that are associated with IT Service Management (ITSM).

- **CSDM implementation stages — Walk**

In the Walk stage, you identify and populate the network infrastructure CIs and applications that your organization's technical teams support.

- **CSDM implementation stages — Run**

In the Run stage, you set up the relationship between a technology and the business that sells and/or consumes the technology.

- **CSDM implementation stages — Fly**

When you reach the Fly stage, you've accomplished all or most of the process of implementing the CSDM framework. The fly stage completes the process.

- **Auto-generate product models for logical CIs**

Use the CSDM Product Model Assignment job to auto-generate a product model record (application model, service model, or software model) for each logical CI that is not yet associated with a product model. Product models are ideal for associating CIs that are parts of a single digital product.

Configure the CSDM Data Foundations dashboard

Use the CSDM Data Foundations dashboard to monitor and evaluate key foundational metrics of the CSDM framework.

Before you begin

For an introduction to the dashboard, see [Viewing the CSDM Data Foundations dashboard](#).

- Before you use the dashboard for the first time, populate the CSDM metrics: Navigate to **All > System Scheduler > Scheduled Jobs** and run the CSDM Get Well Metric Collection job.
- The CSDM Data Foundations dashboard adds the following scheduled jobs that must be running:
 - CSDM Get Well Metric Collection: Calculates and stores details for compliant Cls associated with metrics. Data appears on the dashboard only after the first run of this scheduled job. metrics' scores are stored in the CSDM Data Foundations Metric Scores [sn_getwell_csdm_score] table. The job runs daily by default.
 - CSDM Data Foundations PA Metric Collection: Calculates the total count of non-compliant Cls that are associated with each metric. It also provides trending data over time for the non-compliant Cls associated with metrics.
- Role required: app_service_admin, app_service_user, asset, cmdb_read, itil_admin, portfolio_admin, service_viewer, or technology_service_owner

Procedure

1. Navigate to **All > CSDM > Configuration > CSDM Data Foundations Dashboard**.
2. Select a tab.

The tabs on the dashboard enable you to select your organization's CSDM implementation stage (foundation, crawl, walk, run, and fly). As a result, the reports on each tab display the metrics that are appropriate for the maturity of your CMDB data.

3. Review the reports.
Note the percentages and color-coding in the **Result** column for each metric.

- If the percentage is 100%, the CSDM framework has the information it needs. You don't need to do anything else.
- Otherwise, required information is missing and additional actions are required. Continue with [step 5](#).

Note the metrics on the **Foundation** tab:

Named Product Models with Product Owners

Shows cmdb_model records that meet the following conditions:

- **Status** = in production
- **Name** and **Owner** is not empty

Configuration Item Status Values

Shows the percentage of CIs with default status values.

- 80%: At least 1 to 5 CIs have custom status values.
- 60%: 6 to 10 CIs have custom status values.
- 40%: 11 to 15 CIs have custom status values.
- 20%: 16 or more CIs have custom status values.

To view the default base-system status values, enter `sn_getwell_oob_status_table_field.list` in the navigator **Filter** text box. The Configuration Item Status Values form displays the list of elements and associated tables. Select a table name to see the list of default labels and values.

		Configuration Item Status Values		Search	Label	Search
		Label	Value			
<input type="checkbox"/>	 Retired		7			
<input type="checkbox"/>	 Pending Install		4			
<input type="checkbox"/>	 Installed		1			
<input type="checkbox"/>	 Stolen		8			
<input type="checkbox"/>	 Absent		100			
<input type="checkbox"/>	 In Stock		6			
<input type="checkbox"/>	 Pending Repair		5			
<input type="checkbox"/>	 On Order		2			
<input type="checkbox"/>	 In Maintenance		3			

Business Units with Companies

Shows business unit records where the **Company** field is not empty.

Locations with Parents.

Shows cmdb_ci records that meet the following conditions:

- **Status** = installed
- **Operational status** = operational
- **Location** and **Location.parent** is not empty

4. Select the tiles associated with the foundational metrics to access Performance Analytics widgets.

Performance Analytics widgets are provided by the CSDM PA Metric Collection scheduled job. These widgets provide trending data over time for the non-compliant CIs associated with the metric.



- Ensure that the real-time option is selected () and then select **Show Records** to view the list of CIs.
- Select the **Breakdowns** context menu to view available breakdowns.

5. Scroll to the list of CIs in the **Custom Status Values** related list.

The charts show the number of custom values that have been defined for each element. Click a chart to view custom values that have been defined for the element. This example shows the custom label-value combination for the **Absent** status.

	Element	Label	Value
<input type="checkbox"/>	cmdb_ci	install_status	Absent2 100
<input type="checkbox"/>	cmdb_ci	install_status	Absent 101
<input type="checkbox"/>	service_offering	install_status	Installed by me 1

6. Select a CI to drill down to the form view.

The form view provides the required information. If you don't see the form, you may not have sufficient access privileges. Contact your ServiceNow administrator.

7. When you're finished using the form, select **Update** or **Delete** to return to the list view.
8. Navigate to return to the CSDM Data Foundations dashboard.

Result

The key foundational metric results are available for you to review and analyze.

First activation step — Map existing life cycle data to CSDM standards

The CSDM enforces standard life-cycle states to ensure that assets are tracked accurately over life-cycle transitions. You migrate all life-cycle data across the platform to the CSDM standard.

After you migrate existing data, you will continue to use only standard life-cycle states when creating or updating CIs to gain the following benefits:

- To handle alerts appropriately, Event Management and Operational Intelligence need to know whether a CI is in maintenance life-cycle status.
- To report cost data effectively, Cloud Insights needs to know the state of a CI.
- To generate consistent tasks and workflows, Audit and Compliance need to use standard life cycle values.

You must [activate life cycle migration](#) to migrate legacy life cycle values to the CSDM standard fields and values.

Life Cycle Mapping table

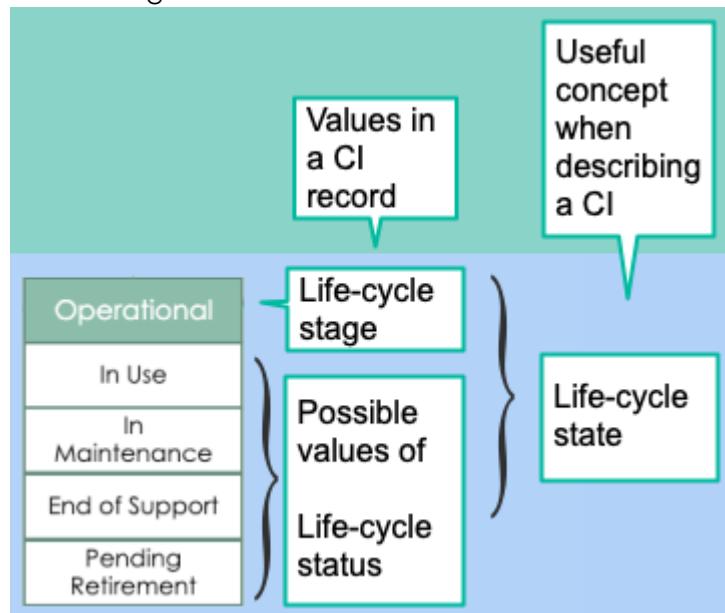
The base system contains the Life Cycle Mapping [life_cycle_mapping] table, which is pre-populated with widely used legacy life cycle

mappings. Each mapping record specifies how to map a legacy life-cycle field's value, based on its table, to the standard CSDM values.

- Life cycle stage (`life_cycle_stage`): The broad life cycle phases that a CI moves through, from inception or procurement to retirement and end of life.
- Life cycle status (`life_cycle_stage_status`): The specific status of a CI within its current life cycle stage.

The table typically contains multiple record entries per class, each entry for a specific legacy life cycle and life-cycle value pair. When there are multiple record entries for a class, the entries are prioritized by importance and likelihood for containing meaningful values for the mapping process.

You can think of the life-cycle state as the combination of two values of an asset or CI: the life-cycle stage and life-cycle status over the CI's life cycle. For example, a hardware CI in the **Operational** stage might change status over time from **In Use** to **In Maintenance** to **End of Support**. A different hardware CI might go from **In Use** to **End of Support** without ever having been in **In Maintenance** status.



Custom life cycle values

In an upgrade scenario, life cycle migration checks for custom legacy life-cycle values that were added in the system. For each custom value, the system adds a record to the Life Cycle Mapping [life_cycle_mapping] table. Those mapping records, however, are incomplete and inactive.

Before you activate life cycle migration, you must edit and activate those records to supply the desired life cycle control to use for mapping.

Specify how to map legacy life-cycle states to CMDB states

Use the Life Cycle Mapping module to specify how your existing life-cycle values should be converted to CSDM life-cycle states. The mapping ensures Now Platform products generate accurate reports for legacy Cls in your environment.

Before you begin

Role required: itil_admin or asset_admin

About this task

In this example, your existing data uses a life-cycle attribute named **Install Status** for hardware Cls. You use the Life cycle mapping form to map the existing **Pending Install** value of the **Install Status** attribute to the **Deploy/Test** life-cycle state in the CMDB.

* Mapping for table **Hardware [cmdb_ci.hardware]**

* Priority **10**

Application

Active

Fields

* Legacy field name	Install Status	Life cycle control	Hardware - Deploy - Test
* Legacy field value	Pending Install	Table	Hardware
Legacy subfield name		Life Cycle Stage	Deploy
Legacy subfield value		Status	Test

Procedure

1. Navigate to **All > CSDM > Life Cycle Mapping**.
2. On the Life Cycle Mappings list view, select **New** and then fill in the Life cycle mappings form.

Field	Description
Mapping for table	Legacy CMDB table and descending tables that this mapping applies to. Applies to a descending table unless there is a mapping configured specifically to the descending table.

Field	Description
Priority	<p>Priority of applying this mapping definition for the table.</p> <p>Priority is used when the life_cycle_mapping table contains multiple entries for a class. The highest priority entry is used first when searching for meaningful legacy values. If the first entry can't be used, the next record in priority is used.</p> <p>Lower numerical values indicate higher priority.</p>
Active	<p>Denotes whether to apply this mapping definition.</p> <p>Deactivation results in lower-priority mappings being used or setting standard life cycle fields to TBD.</p>
Legacy field name	<p>Legacy field in the specified Mapping for table that is currently being used to store a life cycle stage. The value should be used as the source for the life cycle mapping.</p>
Legacy field value	<p>Legacy value in the specified Mapping for table that is currently being used to store life cycle status. The value should be used as the source for the life cycle mapping.</p>

Field	Description
Legacy subfield name	Additional legacy field in the specified Mapping for table that is also used for life cycle management.
Legacy subfield value	Additional legacy value in the specified Mapping for table that is also used for life cycle management.
Life cycle control	Class and life cycle stage and status, that are used as the authoritative source of valid combinations for life cycle mapping.
Table	Standard life cycle table to map the specified Mapping for table to. Setting is based on the selection in Life cycle control .
Lifecycle stage	<p>Standard life cycle stage to map the specified Legacy field name to. Setting is based on the selection in Life cycle control.</p> <p>If there is no match in the <code>life_cycle_mapping</code> table, value is set to TBD.</p>
Lifecycle stage status	Standard life cycle stage value to map the specified Legacy field value to. Setting is based on the selection in Life cycle control .

Field	Description
	If there is no match in the life_cycle_mapping table, value is set to TBD .

3. Select **Submit**.
4. Repeat the process and, when you have mapped all existing life-cycle states, migrate them to the CMDB.
For instructions, see [Activate life cycle migration](#).

Activate life cycle migration

Activate life cycle migration to migrate your mapped legacy custom life-cycle settings to the CSDM standard life-cycle values. The migration script migrates both existing and to incoming data.

Before you begin

Before you activate life cycle migration, navigate to **CSDM > Life Cycle Mapping**. Review the pre-populated mappings in the **Life Cycle Mappings** list view:

- Adjust and add any mappings as needed for your environment.
- Review mappings for any custom legacy life cycle values. Those mappings are incomplete and you must provide the desired standard life cycle control to map to.
- Ensure that all mappings are configured with a life cycle control.
- Ensure that all mappings are activated.

Role required: itil_admin or asset_admin

Procedure

1. Navigate to **CSDM > Life Cycle Mapping**.
2. On the **Life Cycle Mappings** list view, click **Activate**.

Activation can complete only if all mapping records are set to active and are configured with a life cycle control, and all mapping records for custom legacy values are fully configured.

Result

The script performs the following activities:

- One-time bulk mapping of legacy values to the Life cycle stage and Life cycle status fields. The mappings are based on the mapping records in the Life Cycle Mapping table, which contain values, source, and target fields.
-

Change the default false setting for the csdm.lifecycle.migration.activated system property to true. The change activates the Update life cycle from legacy business rule. Future insert or update CI operations will trigger the rule to populate the standard Life cycle stage and Life cycle status fields to ensure that life cycle standards are used continually and consistently.

For example, when creating a hardware CI, and setting the legacy Status and Operational status fields. After saving, the new life cycle standard fields are automatically populated with the matching life cycle standard values based on the corresponding record entry in the Life Cycle Mapping table. If you modify legacy values, the standard fields are automatically updated based on another matching record in the Life Cycle Mapping table.

- Because life cycle migration is a one-time process, the **Activate** button is disabled.

What to do next

After the data has migrated successfully, you can start managing data following the CSDM model:

1. [Activate the CSDM Activation \(com.snc.cmdb.csdm.activation\) plugin.](#)
2. Use the [CMDB Data Manager](#) to centrally govern the life cycle of CIs, in bulk, and in a standard and consistent way.

Second activation step — Activate the CSDM plugin

Activate the CSDM plugin so you can begin implementing the framework.

Before you begin

Important: Before you activate the CSDM plugin, you must map your existing life cycle data to standard CSDM attributes. The mapping enables you to track assets effectively through their life cycle transitions with the [CMDB Data Manager](#). For instructions, see [Migrate to CSDM life cycle standards](#).

- Role required: itil and itil_admin

Procedure

Activate the CSDM Activation plugin: com.snc.cmdb.csdm.activation. For more information, see [Activate a plugin](#).

Third activation step — Migrate existing data to the CSDM framework

You complete several tasks to ensure that your existing application data migrates successfully to the required tables in the CMDB.

Before you begin

Role required: admin

About this task

Keep the following points in mind:

- Some CSDM tables have been introduced recently so you might not be familiar with them. See the documentation for your ServiceNow product to learn about unfamiliar tables.

- You can continue to use customized or non-conforming CMDB tables. If you do so, however, you may not get the full benefit of your ServiceNow products.
- Be sure to use the migration tools that are described in [CSDM migration tools](#).

Manage the attributes that you are using. Rationalize your custom attributes. Use the following guidelines to decide whether you really need to keep all customizations:

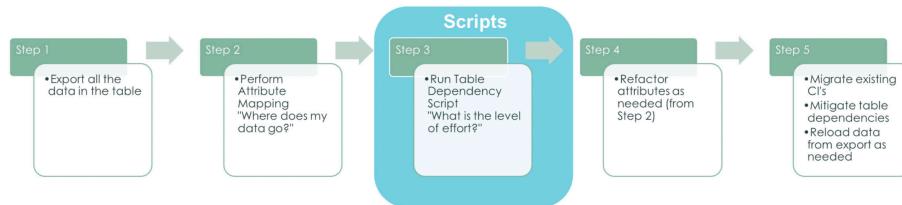
- Best Practice: The custom attribute doesn't have a related base-system attribute but you need to use it.
- Keep: The custom attribute doesn't have a related base-system attribute but it's required for a unique use case.
- Refactor: The custom attribute does have a base-system attribute or a capability that can be migrated.
- Do Not Need: The customization is no longer needed. Delete the attributes that you don't use or use only rarely. Consider deleting attributes if there's a better way to address a use case.

Consider related dependencies. Moving configuration items (CIs) to a new table doesn't automatically move related dependencies. To identify related dependencies, use the scripts described in [Migrating into CSDM identifying table dependencies](#) (available on the ServiceNow Community).

Important: The script doesn't move your data or their dependencies. It only identifies the dependencies. You refactor data and dependencies as part of the migration.

After running the scripts and evaluating the data, you will have a better idea of the effort required to migrate your data. Decide whether you need all referenced reports, rules, and scripts. Then decide what you want to migrate and make a migration plan.

Migration workflow



Procedure

1. Back up your data.

Export your data (with all attributes) to Excel and keep the file in a secure location. Have a contingency plan in case issues arise.

2. Map the attributes.

Identify the table where the data should go. Make sure that destination table has the required base-system attributes. Rationalize your custom attributes. Decide which customizations you will keep.

3. Move CIs from existing classes to CMDB classes.

Note: Don't forget about non-conforming tables and their dependencies. You could have hundreds of reports, business rules, scripts, table references, and more that need the data in your non-conforming tables.

Moving CIs to a new table does not automatically move reports, business rules, and so on. As described in the following steps, the fix script identifies dependencies that you need to refactor. You can [download the fix script](#) from the ServiceNow Community.

4. Refactor the attributes.

Solidify the data model and get the data ready for migration.

Make sure you have completed the attribute mapping-related tasks described in earlier steps. Follow the guidelines and refactor your data as needed.

5. Migrate the data.

Note: Make sure that you have a valid and recent backup. Make another backup if necessary. During migration, you lose all customized or base-system attributes that are not in the same table hierarchy.

Keep these points in mind as you proceed:

- Migrate your CIs to the new class to move the CI and all its related objects, incidents, and changes to the new table.
 - Start with a few CIs and increase the number when you feel comfortable.
6. Remediate your table dependencies:
- a. Modify reports to use the new table.
 - b. Migrate business rules and scripts as needed.
 - c. Update table references as needed.
7. Reload data into the new attributes using the backup that you made earlier.
8. Validate all data and dependencies.

Result

You've successfully migrated your application to the CSDM framework, and your data is in the required CMDB locations.

- [CSDM migration tools](#)

Migration tools simplify the process of migrating your data to the CSDM framework.

Migration tools simplify the process of migrating your data to the CSDM framework.

Data migration script

Use the [fix script](#) from the ServiceNow Community to identify table dependencies.

CMDB Data Foundations dashboard

Navigate to **All > Configuration** and then select **CMDB Data Foundations Dashboard**.

- CMDB dashboard: Each tab provides key metrics that evaluate configuration and customizations in the CMDB.
- The metrics provide visible results of evaluation. Color codes and weighted priority help with planning.
- Each metric includes a link to a remediation playbook with background and plays for remediation.

See [Monitor system foundations in the CSDM and the CMDB Data Foundations Dashboards \(2.2.1\)](#).

CSDM Data Foundations dashboard

Navigate to **All > Configuration** and then select **CSDM Data Foundations Dashboard**.

The CSDM dashboard focuses on key data elements to support you in implementing the CSDM framework. The tabs on the dashboard enable you to select your organization's CSDM implementation stage (foundation, crawl, walk, run, and fly). As a result, the reports on each tab display the metrics that are appropriate for the maturity of your CMDB data. See [Viewing the CSDM Data Foundations dashboard](#).

CSDM implementation stages — Foundation

In the Foundation stage of implementing the CSDM framework, you prepare the referential data that enables accurate reporting to support good business decisions. Use the base-system tables when you begin implementing the CSDM to derive the highest value from your ServiceNow products and the Now Platform.

Benefits of preparing the data in the Foundation stage

The basis of any good data model is the foundational data that is referenced throughout the model.

- The base-system tables act as the foundation for many ServiceNow products.

- The tables help your company align with reporting requirements early to expedite the value you get from the CSDM. You can reduce or eliminate costly rework tasks needed to align with reporting requirements.

Tables that you work on during the Foundation stage



Business Process table [cmdb_ci_business_process]

A business process has a well-defined start and finish. Examples of business processes in the banking industry are the customer onboarding process and the credit check process. Each business process can have levels of criticality and impact. Business processes are stored in the cmdb_ci_business_process table.

Contract table [ast_contract]

The Contract table identifies binding agreements between two parties. When you populate services provided by vendors into the CMDB, consider the role that contracts play when evaluating service level agreements (SLAs).

Product model table [model_id]

The Product model table [model_id] identifies the unique types of products your organization develops or consumes. When you group assets and CIs by product model, you unify and relate CIs that are part of the same digital product and portfolios of products. Grouping assets and CIs by product model can help you plan projects, monitor costs, and rationalize your data. Discovery can populate hardware product models after they're operational, but other types of product models require planning from product owners.

Use the CSDM Product Model Assignment job to auto-generate a product model record (application model, service model, or software model) for each logical CI that is not yet associated with a product

model. Product models are ideal for associating CIs that are parts of a single digital product. See [Auto-generate product models for logical CIs](#).

CMDB Group table [cmdb_ci_query_based_service]

The CMDB Group table identifies a collection of CIs based on the results of saved Query Builder queries, encoded queries, or manual entries. CMDB groups are critical elements of Dynamic CI groups and the strategic management of CIs. Decide early how you want to report CI information and how you want to monitor CIs. These decisions affect how you create CMDB groups.

Location table [cmn_location]

The Location table uniquely identifies geographic locations. You can create a hierarchy of location data using the Parent attribute. The hierarchy might include entries that match your reporting requirements. For example, you could populate the location table as follows:

Your organization's location attributes



To include more detail in reports, you could extend the Location table to include floors, rooms, and even datacenters. With hierarchy capabilities, trusted source data, and your requirements in hand, you can create locations that support your future reporting needs.

Group table [sys_user_group]

The Group table identifies sets of users that share a common purpose. Groups may perform tasks such as approving change requests, resolving incidents, receiving email notifications, or performing work order tasks. Groups also use the referential data in the CMDB to identify how CIs are managed (for example, the Managed by group) and supported (for example, the Support group). Any business rules, assignment rules, system roles, or attributes that refer to a group automatically apply to all group members.

User table [sys_user]

The User table identifies the persons and applications that have access to your ServiceNow instance. You can organize users into groups that are associated with the Company, Business Unit, and Department tables.

Organizational structure

Organization structure tables identify internal business structures and external customers, manufacturers, and vendors.

Company table [core_company]

The Company table is populated with the legal entities of companies. Entities can be either internal (your organization) or external. You can use the Parent attribute to build a hierarchy. Consider the legal entities that you need for reporting when the CMDB is populated.

- Internal entries should focus on a hierarchy of legal entities rather than a hierarchy of business units within a legal entity.
- External entries are identified by a True or False flag. The Customer flag identifies your external customers.

The Manufacturer flag identifies companies that create products that you consume. An internal organization might be a manufacturer.

- The Vendor flag identifies organizations that provide products that you purchase. An internal organization might be a vendor.

Business Unit table [business_unit]

The hierarchy of your business is populated in the Business Unit table with a reference to the parent company. A business unit is a part of your organization that is responsible for specific operations, such as finance, human resources (HR), or IT. A hierarchy within a business unit is common. For large multinational organizations, you may have business units that identify independent regional operations and the specific operations within the region.

Department table [cmn_department]

The Department table includes a finer level of detail about a business unit. The Department table gives you another way to categorize users, groups, assets, and Cls.

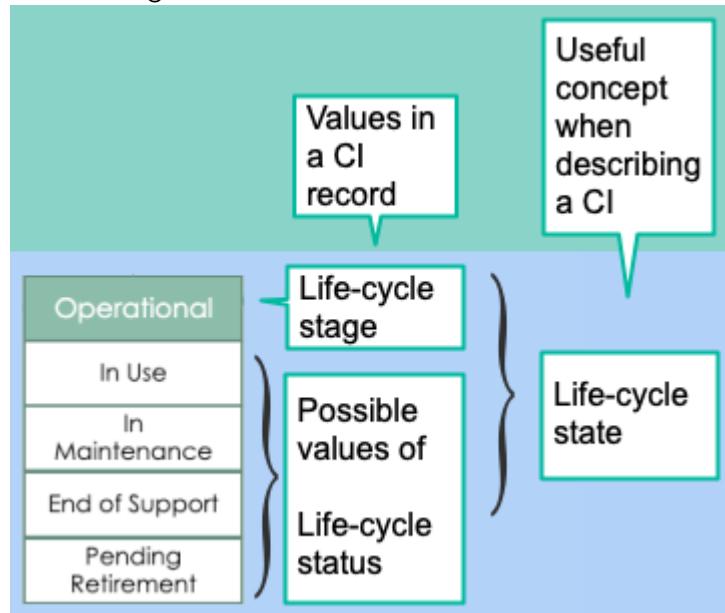
Life-cycle tables

Life-cycle states track the life cycles for products, assets, contracts, Cls, locations, and other objects. Using the standard life-cycle values consistently helps you to track objects through their transitions over time.

Reporting can therefore accurately reflect the actual states of CIs: usage, availability, end of support, and so on.

Note: Based on the type of item, the [life_cycle_control] table controls which life-cycle stage values are available for each life-cycle stage.

You can think of the life-cycle state as the combination of two values of an asset or CI: the life-cycle stage and life-cycle status over the CI's life cycle. For example, a hardware CI in the **Operational** stage might change status over time from **In Use** to **In Maintenance** to **End of Support**. A different hardware CI might go from **In Use** to **End of Support** without ever having been in **In Maintenance** status.



When you enable the CSDM framework, you can start using the Life Cycle Stage and Life Cycle Stage Status fields to track an asset's life cycle. To use the fields, follow the procedure described in [Second activation step — Activate the CSDM plugin](#). The following assets can use life-cycle states:

- Product life cycle
- Hardware life cycle
- Logical life cycle

- Document life cycle
- Location life cycle

Watch the ServiceNow Community video: [CSDM V4 product and life cycle discussion](#)

CSDM implementation stages — Crawl

In the Crawl stage, you work on base-system CMDB tables that are associated with IT Service Management (ITSM).

Benefits of the operations that you perform in the Crawl stage

- The operations provide the minimum CMDB support requirements for Incident Management and Change Management.
- Setting up APM is faster because your business application data is in the right place in the CMDB.
- The operations build the foundation for using DevOps because your SDLC component data is populated and ready to relate to your applications.
- Service Mapping is ready to use for mapping entry points because your application service data is populated.
- The operations build the foundation for using TPM risk details, a capability of APM.

The operations prepare you to manage and monitor the life cycles and versions of the underlying technologies of the business applications in your enterprise.

The data enables you to identify outdated or at-risk software using APM, Service Mapping and Software Asset Management (SAM) Professional.

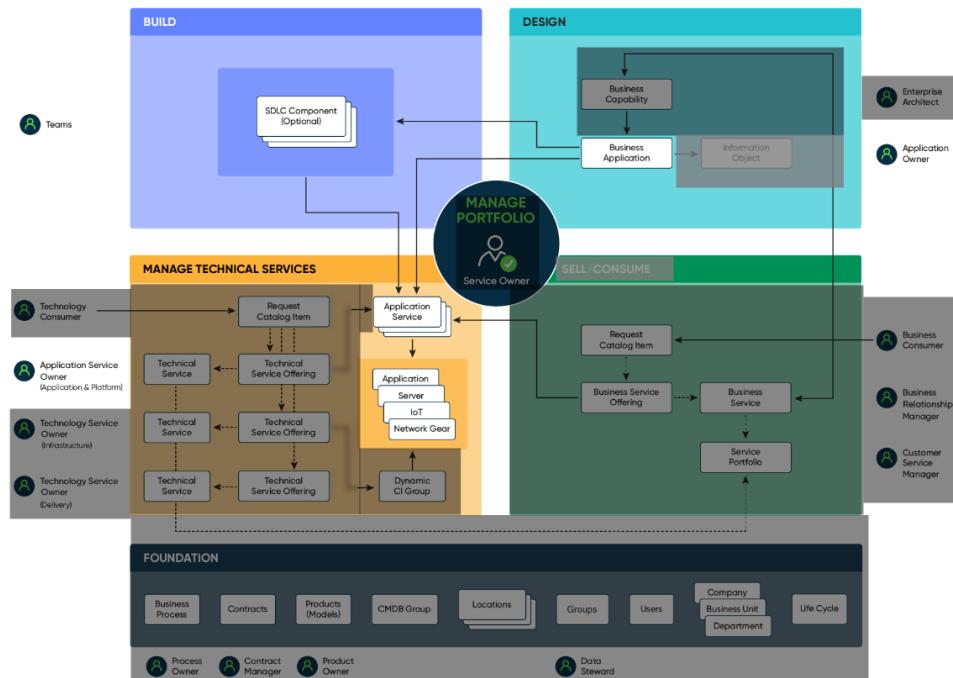
Tables that you work on during the Crawl stage

Important: Future products and product enhancements depend on the data that you prepare in each of the tables.

During this stage, you work on the following base-system CMDB tables:

- Business Application table [cmdb_ci_business_app]
- Mapped Application Service table [cmdb_ci_service_discovered]
- Application table [cmdb_ci_appl] (discoverable)
- Server/host (discoverable)

Note: Some of the classes that you implement in this stage are logical CIs. Logical CIs aren't created through Discovery, so their **Model ID** values might not refer to product model (application model, service model, or software model) records. To help you to migrate to a product-centric management paradigm, each instance of a logical CI should be associated with a product model. See [Auto-generate product models for logical CIs](#).



Start by focusing on applications and the application-related data in these areas and tables:

Business Application table [cmdb_ci_business_app]

A business application is a base-system CMDB table that stores your inventory, application portfolio, and their metadata.

Because this table is not an operational configuration item (CI), it is not used by ITSM Incident Management, Problem Management, or Change Management processes.

SDLC Component table [cmdb_ci_sdlc_component]

SDLC component CI records in the SDLC Component table [cmdb_ci_sdlc_component] enable the DevOps product to provide enhanced capabilities for visualizing and managing your application development pipeline.

This table represents the software part or element of a larger whole for applications and infrastructure. Related material may serve as representative of developmental details. It can be used if you need to identify the stratification of a business application or digital product.

Note: This is not an operational CI and cannot be used in incident, problem, and change.

Application Service table [cmdb_ci_service_auto]

The application service is typically the system that the caller identifies when they report an issue with an application.

A mapped application service is a base-system CMDB table that identifies the related business application in use. The application service ties all the elements of the CSDM together where applications are present.

You may have several application services representing each deployment based on the environment (development, QA, production) and location or geography (North America, Asia Pacific).

Because application services are logical in nature, they should use the Logical life cycle states. Application services follow the same life cycle guidance as any other logical CI.

Application table [cmdb_ci_appl]

An application is a base-system CMDB table that represents the discoverable instance of an application: code related to a process in use on a host. This table isn't an inventory of your applications. Because of the high level of complexity involved, don't try to manually populate the application table. Discovery creates and maintains this table.

Important: The application table [cmdb_ci_appl] isn't an inventory or portfolio of your applications. Don't make the mistake of storing managed application details in the application table. Those details (inventory or application portfolio objects) belong in the business application table (as documented in [Design domain of the CSDM framework](#)).

The application may be identified as the root cause of an incident. However, if you're not using Event Management, the application may not be the initial cause.

If you're using Discovery, applications are automatically related to their host, which provides an impact hierarchy from server-to-host applications.

CSDM implementation stages — Walk

In the Walk stage, you identify and populate the network infrastructure CIs and applications that your organization's technical teams support.

Benefits of the operations that you perform in the Walk stage

Managing discovered infrastructure CIs

The operations facilitate managing the discovered infrastructure CIs. You might manually managing the metadata on these CIs, such as support group and technical approval group. By identifying the technical service offering that manages these CIs, you can:

- Configure ServiceNow to populate and synchronize this metadata onto the related child objects.
- Eliminate the manual effort of maintaining the metadata on thousands of CIs.

View supported CIs

The operations establish a view of the CIs that your organization's technical teams support.

You can see the specific support assignments, which you can change as needed based on your support structure, operational-level agreements (OLAs), and commitments.

Also, this view enables you to formalize for your process for supporting applications and technology owners.

Prepare for Service Portfolio Management

The operations build the foundation for using Service Portfolio Management (Service Portfolio Management).

You can start using Service Portfolio Management more quickly because your service data are in the right place.

Use the Request Catalog

The operations enable you to order technology service offerings through the [Request Catalog](#). You can also automate ordering some offerings to enhance the request workflow and update or create related CIs.

Note: The Request Catalog is not a CMDB table.

Prepare for ITOM products

The operations build the foundation for Information Technology Operations Management (ITOM) products, such as Service Mapping and Discovery.

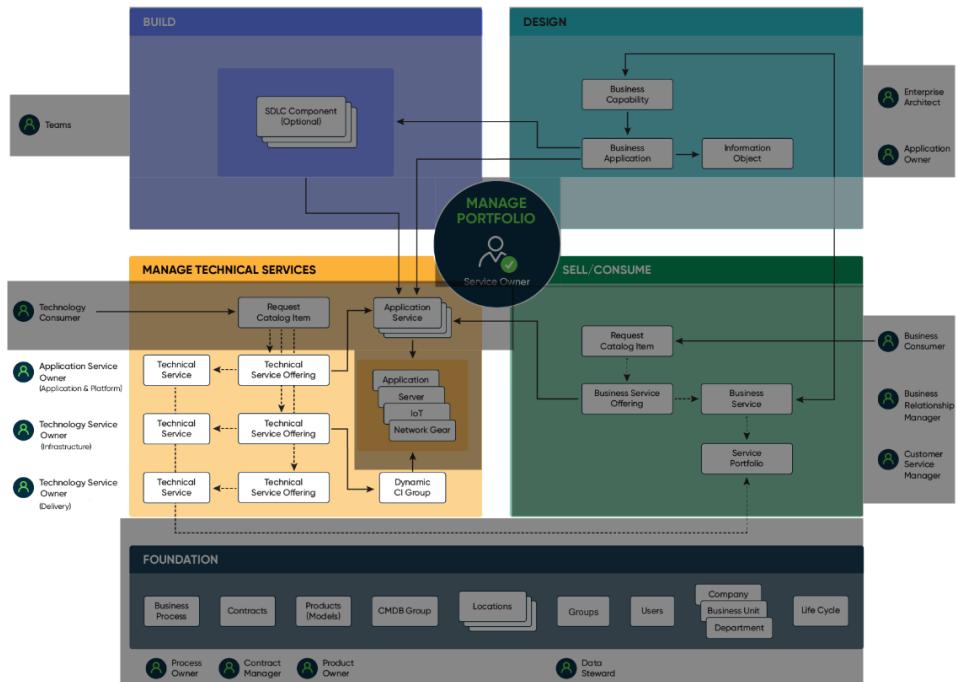
Enable Automation

The operations enable more automated methods of grouping CIs for identification and management by Technical Service Offerings.

Tables that you work on during the Walk stage

The walk stage includes base-system CMDB tables that identify the technology provider.

Note: Some of the classes that you implement in this stage are logical CIs. Logical CIs aren't created through Discovery, so their **Model ID** values might not refer to product model (application model, service model, or software model) records. To help you to migrate to a product-centric management paradigm, each instance of a logical CI should be associated with a product model. See [Auto-generate product models for logical CIs](#).



Technical service table [cmdb_ci_service_technical], or [cmdb_query_based_services] for Event Management

The Technical service table has a service classification of "technical service". This base-system CMDB table identifies the provider of the technology that your business consumes.

Technology service offering table [service_offering]

A Technical service offering is a service offering with a service classification of "technical service". Technical service offerings may be further divided as follows:

- Location and geography
- Environment (production or non-production)
- Pricing
- Availability
- Support group (for Incident Management)
- Technical approval group (for Change Management)
- Packaging options (commitments)

The technical service offering comes from the service, based on how the parent serves a specific technical need. Every operational technical service must be associated with at least one technical service offering.

Note: Not all technical service offerings have to be related to applications or infrastructure Cls. Managed Service Providers may provide technical service offerings.

Dynamic CI group table [cmdb_ci_query_based_service]

A Dynamic CI group is a collection of Cls based on the results of saved Query Builder queries, encoded queries, or manual entries. Query Builder is described in [Querying the CMDB](#). For more information about Dynamic CI groups and how you can use them, see [Manage Technical Services domain of the CSDM framework](#).

CSDM implementation stages — Run

In the Run stage, you set up the relationship between a technology and the business that sells and/or consumes the technology.

ITSM considerations during the Run stage

When you use ITSM, you must understand the impact that a technology can have on your business. For example, your business may:

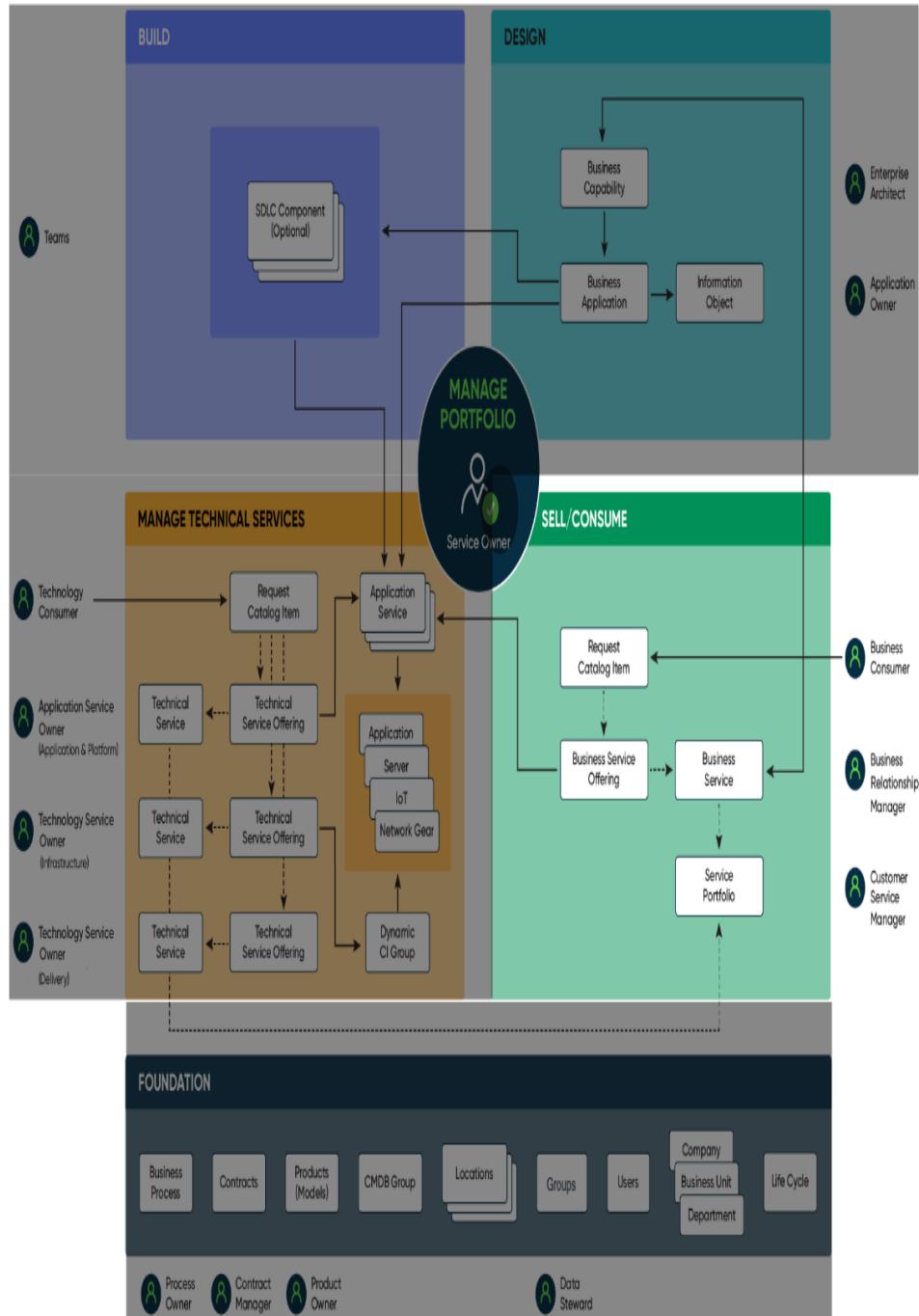
- Consume the technology.
- Sell the technology (as is the case with Customer Service Management).

- Both sell the technology and consume it.

Benefits of the operations that you perform in the Run stage

- Run-stage operations ensure impact assessment for Incident Management and Change Management. Within an incident or change, you can identify the impacted business, assuming relationships exist between the selected CI and the impacted businesses.
- Run-stage operations provide a foundation for using Service Portfolio Management in the Digital Portfolio Management (DPM). Service owners can monitor service portfolios and understand service-related information including service trends, improvement initiatives, service performance, and outage monitoring.
- Run-stage operations provide a foundation for ITSM capabilities. This foundation populates the related “Subscribe by” table on a service offering to identify the business and subscribers affected. Business service offerings can identify subscribers by user, company, location, department, and group.

Tables that you work on during the Run stage



Note: Some of the classes that you implement in this stage are logical CIs. Logical CIs aren't created through Discovery, so their **Model ID** values might not refer to product model (application model, service model, or software model) records. To help you to migrate to a product-centric management paradigm, each instance of a logical CI should be associated with a product model. See [Auto-generate product models for logical CIs](#).

Business service portfolio table [service_portfolio]

Note: The Business service portfolio is not a CMDB table.

A business service portfolio is not a CMDB table. A business service portfolio is a hierarchical collection of business services (products and services) that define a business objective.

Business service table [cmdb_ci_service_business]

The business service table is a base-system CMDB table. This table identifies a business objective that uses (and depends on) the infrastructure that technology uses.

This dependency means that the business service must sell or consume that infrastructure.

Business service offering table [service_offering] (service offering classified as a "business service")

Business service offerings are the starting point for configuring Service Portfolio Management. Business service offerings consist of one or more service commitments. These service commitments uniquely define the level of service in terms of availability, scope, pricing, and other factors.

The business service offering comes from the service. The business service offering is fine-tuned based on how the parent serves a specific technical need.

Every business service should have at least one business service offering.

- [Catalogs and catalog items](#)

A catalog (sometimes called a request catalog or service request catalog) is a set of business and technical products, services, service commitment options, and offerings that users can order on a self-service basis. You can manage a catalog to present your available products and services to users as catalog items.

A catalog (sometimes called a request catalog or service request catalog) is a set of business and technical products, services, service commitment options, and offerings that users can order on a self-service basis. You can manage a catalog to present your available products and services to users as catalog items.

Catalogs (such as the Human Resources [HR] service catalog) help manage services that a user can access. Catalogs contain catalog items and are the starting point for accessing available services.

Catalog items

A catalog item is an item or a service that you can request from the catalog. A service (for example, the employee onboarding catalog) can offer multiple catalog items. Catalog items are listed on the service portal and are available to the users who need them (either through subscription or job responsibility). Each catalog item is linked to one service offering.

CSDM implementation stages — Fly

When you reach the Fly stage, you've accomplished all or most of the process of implementing the CSDM framework. The fly stage completes the process.

Benefits of the operations that you perform in the Fly stage

The Fly stage is a foundation for using APM capabilities

You can use APM capabilities to rationalize your business applications. Ask questions such as the following:

- Are you spending too much on your business capabilities?
- Are you spending too little on your business capabilities?
- Should you increase the amount you spend on emerging business capabilities?

The Fly stage is a foundation for using APM with Service Portfolio Management capabilities

You can use APM with Service Portfolio Management capabilities to rationalize your business services and related offerings. Ask questions such as the following:

- Are you spending too much or too little on services?
- Are you spending too little or too little on services?
- Are they the right services compared to emerging capabilities?

The Fly stage is a foundation for using ITSM capabilities

Starting with the New York release, you can use the [Request Catalog](#) to relate a service offering to a catalog item. You can also enhance the request workflow to automatically populate the "Subscribe by" table.

Note: The Request Catalog is not a CMDB table.

The Fly stage can help you to manage business services

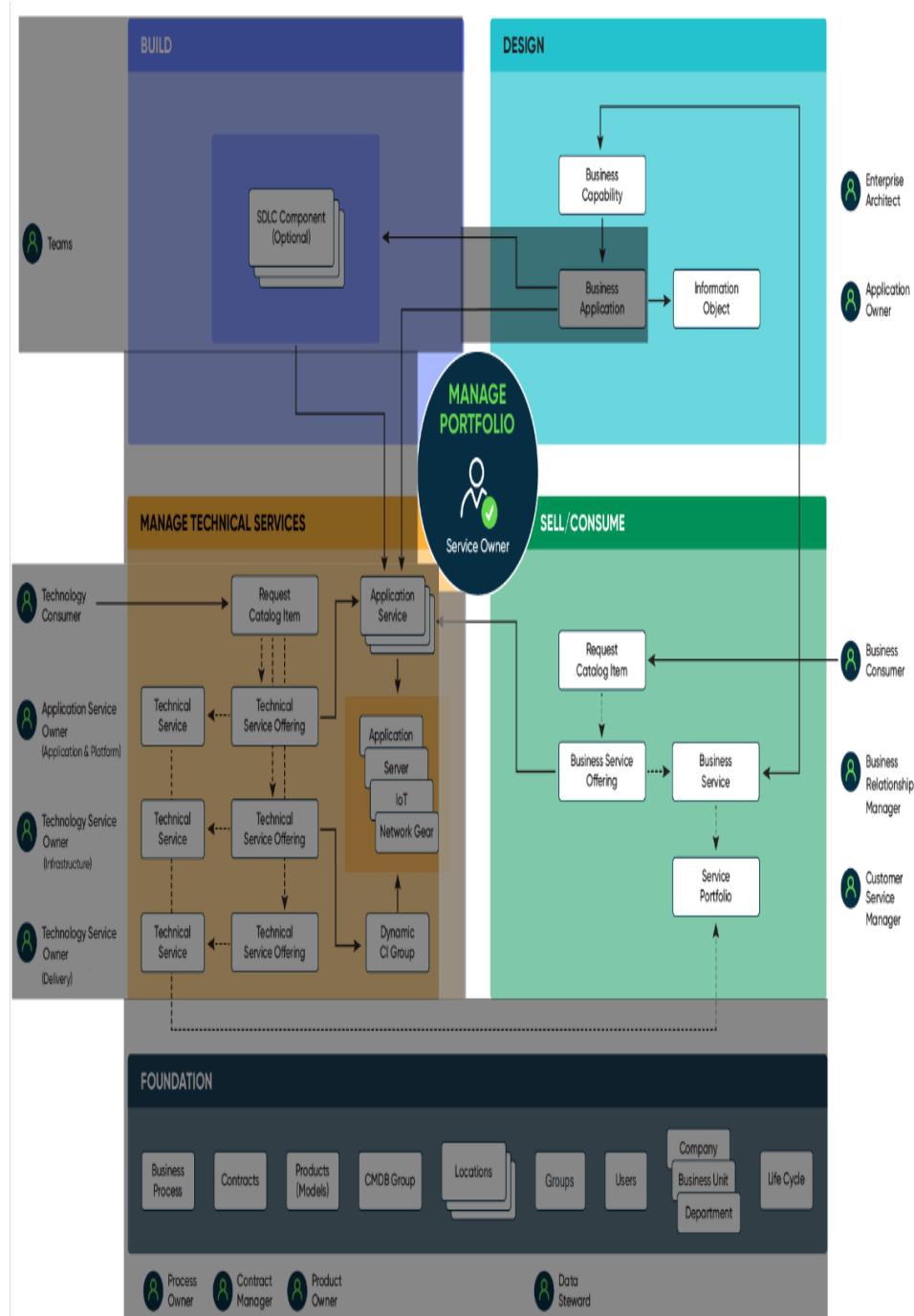
If your environment has a combination of CIs from each of the CSDM domains, this stage provides away to manage business services.

The Fly stage is a way to identify the types of data that may be contained in or used by your business applications

The information object table helps you see what's in your information portfolio.

Tables that you work on during the Fly stage

You reach the fly stage after you have accomplished all or most of the earlier stages.



Note: Some of the classes that you implement in this stage are logical CIs. Logical CIs aren't created through Discovery, so their **Model ID** values might not refer to product model (application model, service model, or software model) records. To help you to migrate to a product-centric management paradigm, each instance of a logical CI should be associated with a product model. See [Auto-generate product models for logical CIs](#).

The fly stage completes the remaining aspects of CSDM framework:

Business capability table [cmdb_ci_business_capability]

A business capability is a high-level capability that supports a business model or fulfills a mission for your organization.

Information object table [cmdb_ci_information_object]

You capture asset information as information objects. You can connect the information objects to your business applications to create an application portfolio that you can use at any time.

You can use the Information Object table to identify the types of data that a business application uses, including highly sensitive data such as:

- Personally Identifiable Information (PII)
- Payment Card Industry Data Security Standard (PCI DSS) data
- Health Insurance Portability and Accountability Act (HIPAA) data

Information objects are part of the information portfolio. The information portfolio links to the following data:

- Data Domains: Total number of records in the Data Domain table [sn_apm_data_domain].
- Information Objects: Total number of records in the Information Object table [cmdb_ci_information_object].
- Database Instances: Total number of records in the Database Instance table [cmdb_ci_db_instance].
- Database Catalogs: Total number of records in the Database Catalog table [cmdb_ci_db_catalog].

Important: You might need to implement the Information object table [cmdb_ci_information_object] as part of an earlier stage. Your business requirements determine the right stage for implementing the table.

Request catalog

Users request services through the [Request Catalog](#). The Request Catalog is not a CMDB table.

The fly stage includes these components:

- Business service portfolio table [service_portfolio]
- Business service table [cmdb_ci_service_business]; (service classified as a "business service")
- Business service offering table [service_offering]; (service offering classified as a "business service")

Auto-generate product models for logical CIs

Use the CSDM Product Model Assignment job to auto-generate a product model record (application model, service model, or software model) for each logical CI that is not yet associated with a product model. Product models are ideal for associating CIs that are parts of a single digital product.

Before you begin

Users with read access to modified CI records can view the new product models.

Role required: admin

About this task

Application, service, and software class instance CIs are not created through Discovery, so their **Model ID [model_id]** values might not refer to product model records. To assist you in migrating to a product-centric management paradigm, each instance of a logical CI should be associated with a product model. The CSDM Product Model Assignment

job operates on the following classes and uses the name of the class instance as the name of the new product model.

- Service Offering
- Technical Service
- Application Service: The script adds the version (as it relates to the Business Application) to the associated software model name. For example, the software model for the MyAppService application service CI might be **MyAppService - version: 2.1**.
- Business Service
- Business Application

The CSDM Product Model Assignment job calls the CSDMMModelUtil script. The script performs these actions for each instance of the supported CSDM classes that does not refer to a product model:

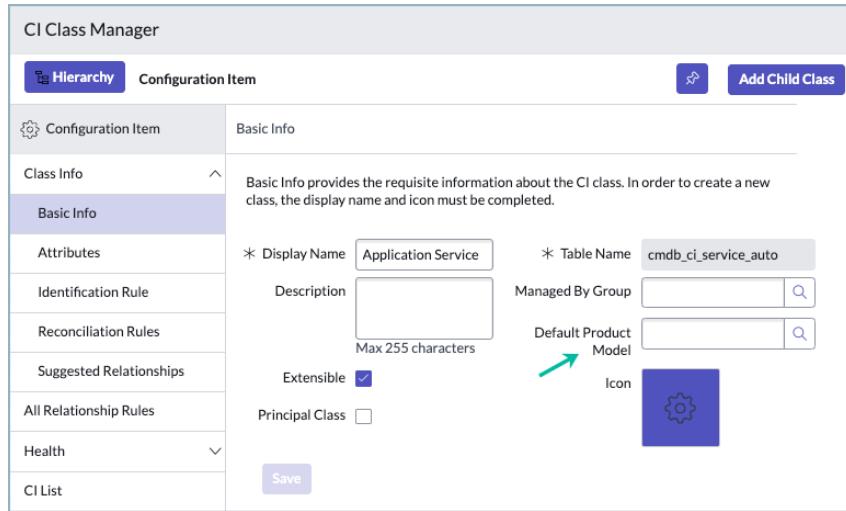
1. Create a new application model, service model, or software model record with the same name as the CI. If the requisite info for generating the name value does not appear in the CI, the script uses the default value that you specified.
2. For the CI, add a reference to the new product model in the **Model ID [model_id]** field.

Operation of the script:

- The script observes the access rules of the CMDB admin that runs the script.
- If an auto-generated product model record for a class instance CI would be identical to an existing record, then the existing record is used for that CI.
- If your data includes CIs with identical names (this is actually an error), then the resulting model_id values might conflict. Validate the resulting model_ids after running the script.

Procedure

1. Specify the default name to use if the name / version values of a CI are insufficient to auto-generate the name for the new product model.
 - a. Navigate to **Configuration > CI Class Manager** and then select **Open Hierarchy**.
 - b. Navigate to each of the supported classes in turn (Application Service, Business Service, and so on).
 - c. On the **Basic Info** page for the class, enter the appropriate name in the **Default Product Model** field.



2. Run the CSDM Product Model Assignment job.
 - a. Navigate to **System Scheduler > Scheduled Jobs > Scheduled Jobs**.
 - b. Search for and open the CSDM Product Model Assignment job.
 - c. Select **Execute Now**.
The script runs and generates the product models.
3. Review the created product models to ensure that the new **Model ID** [**model_id**] values are correct.
Use:

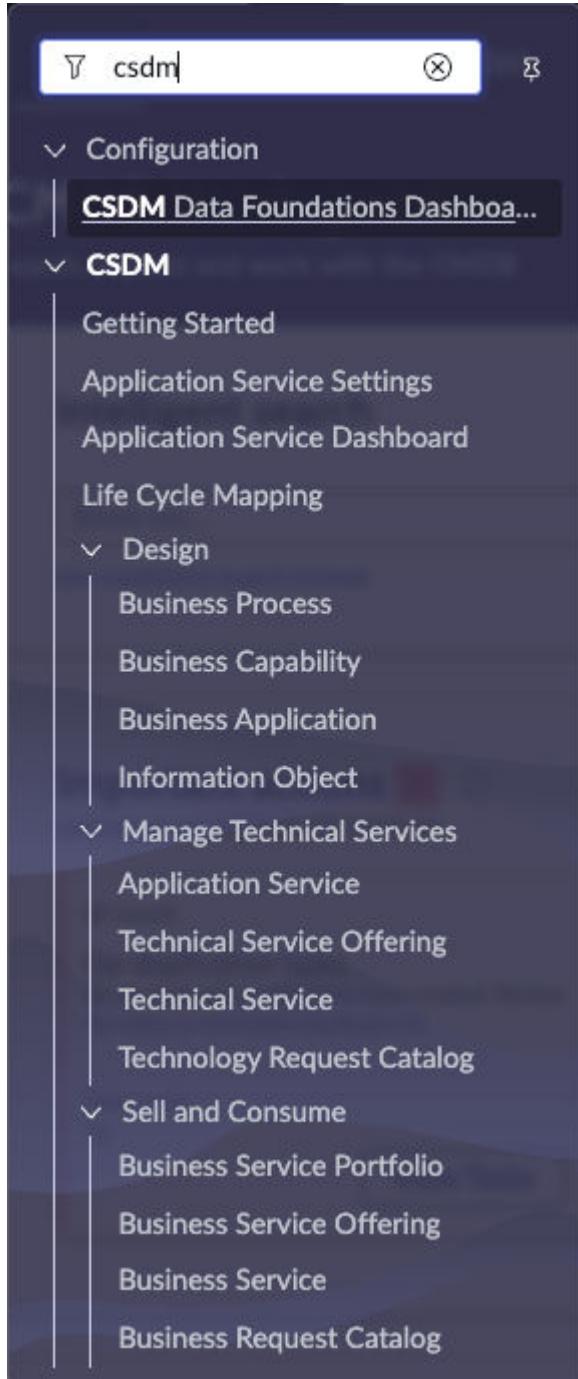
- **All > Product Catalog > Product Models > Application Models**
 - **All > Product Catalog > Product Models > Service Models**
 - **All > Product Catalog > Product Models > Software Models**
4. Run the job whenever you want to create product model settings for new class instances.

Managing the CSDM framework

The CSDM is the data framework that you follow when you set up ServiceNow products and applications. You adhere to the CSDM guidelines when you define configuration items (CIs) and relationships between CIs in the CMDB. This process ensures that your data resides in the appropriate CMDB tables for maximum value from your Now Platform applications.

CSDM modules

Navigate to the modules that assist you in implementing and managing the CSDM domains and their components.



CSDM Data Foundations Dashboard

The CSDM Data Foundations dashboard displays key CSDM indicators on a single page to help you get the full benefit from your Now Platform products. See [Viewing the CSDM Data Foundations dashboard](#).

Getting Started

Select **Getting Started** to open the library of CSDM user documentation — the documentation you are viewing now.

Application Service Settings

Use the Application Service Settings module to specify the attributes and relationships that are required when a user creates an application service. See [Specifying attributes and relationships for Application Services](#).

Application Service Dashboard

The Application Service dashboard enables you to monitor and manage application services to ensure that application services are fully configured and are populated in the CMDB. See [Monitoring and managing application services](#).

Life Cycle Mapping

Use the Life Cycle Mapping module to specify how your existing life-cycle values should be converted to CSDM life-cycle states. The mapping ensures Now Platform products generate accurate reports for legacy CIs in your environment. See [First activation step — Map existing life cycle data to CSDM standards](#).

Design

Work in the tables that are referenced in the Design domain of the CSDM. See [Design domain of the CSDM framework](#).

Manage Technical Services

Work in the tables that are referenced in the Manage Technical Services domain of the CSDM. See [Manage Technical Services domain of the CSDM framework](#).

Sell and Consume

Work in the tables that are referenced in the Sell/Consume domain of the CSDM. See [Sell/Consume domain of the CSDM framework](#).

Synchronize data for 'Managed by' and 'Change' groups

Synchronizing group assignment attributes

Set the group for a CI or an entire class of CIs

Synchronize data using a technical service offering

Implement CSDM for your application

Application Portfolio Management product view

Business Continuity Management product view

Change Management product view

Customer Service Management product view

DevOps Config product view

Incident Management product view

Operational Technology product view

ITOM Health product view

ITOM Visibility product view

Problem Management product view

Service Catalog product view

- [Viewing the CSDM Data Foundations dashboard](#)

The CSDM Data Foundations dashboard displays key CSDM indicators on a single page to help you get the full benefit from your Now Platform products.

- [Specifying attributes and relationships for Application Services](#)

Use the Application Service Settings module to specify the attributes and relationships that are required when a user creates an application service.

- [Monitoring and managing application services](#)

The Application Service dashboard enables you to monitor and manage application services to ensure that application services are fully configured and are populated in the CMDB.

- [Synchronizing group assignment attributes](#)

To empower a particular user group to manage a collection of CIs or CI classes, set group assignment attributes through the technical service offering or the CI Class Manager. The operation synchronizes the group attribute data across all CIs that belong to the specified CI class or groups of CIs.

Viewing the CSDM Data Foundations dashboard

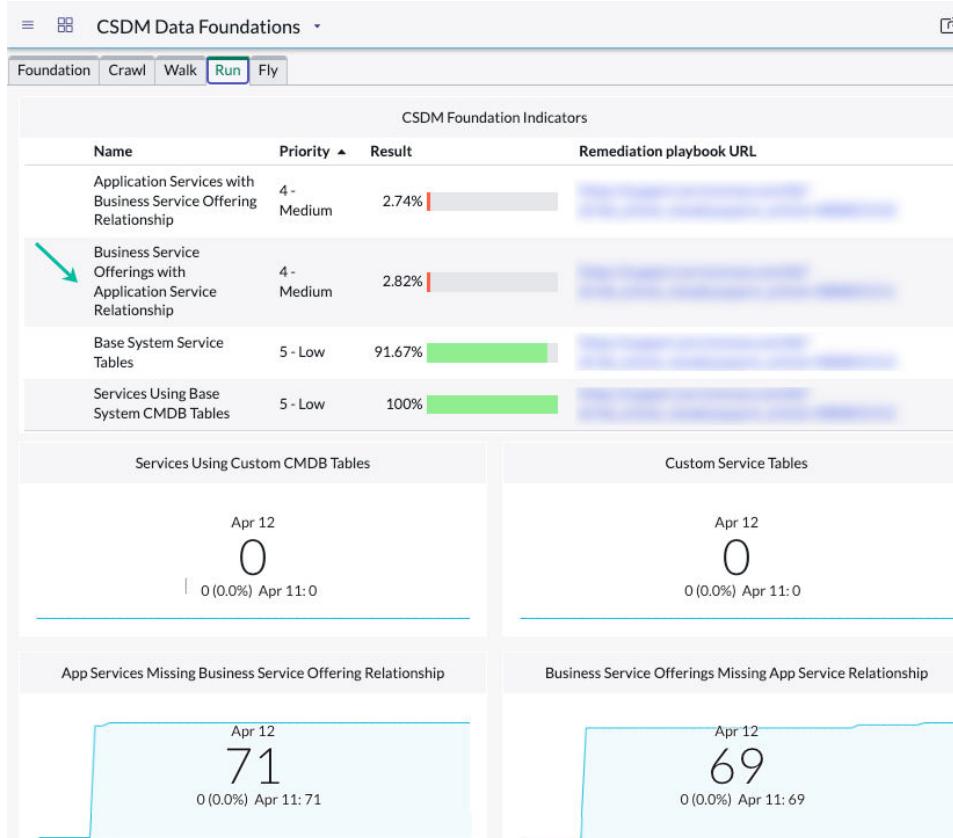
The CSDM Data Foundations dashboard displays key CSDM indicators on a single page to help you get the full benefit from your Now Platform products.

CSDM Data Foundations dashboard

The tabs on the dashboard enable you to select your organization's CSDM implementation stage (foundation, crawl, walk, run, and fly). As a result, the reports on each tab display the metrics that are appropriate for the maturity of your CMDB data. The label on each report identifies in plain language the metric being displayed.

Select **All > CSDM Data Foundations Dashboard** to open the dashboard.

In this example, a report on the **Run** tab indicates that several business service offerings don't have the required relationships to application services. With this knowledge, Service owners can add the relationships to ensure that customer service agents get complete information on the upstream impacts of applications that are down.



- The **Priority** value is the product of the weight of the metric and the severity of the actual score. Priority ranges from 1 — Critical (the highest priority), to 5 — Low (the lowest priority).

-

The **Result** column displays a color-coded bar showing the percentage of CIs or the measured item that are in compliance for the key foundational metric.

- Red: 0–50% are in compliance.
- Yellow: 50–90% are in compliance.
- Green: More than 90% are in compliance.

- The **Remediation playbook URL** column displays links to knowledge articles in Now Support with instructions for bringing the CIs into compliance. Use your Now Support credentials to access the knowledge article.

Note: Starting with the CSDM Data Foundations dashboard v2.2, the following metrics no longer appear on the **Fly** tab:

- Information objects missing an app service relationship
- Catalog request items related to service offerings

Further information

- For an introduction, watch the [ServiceNow Data Foundations Dashboards for CSDM and CMDB](#) video.
- See [Configure the CSDM Data Foundations dashboard and Monitor system foundations in the CSDM and the CMDB Data Foundations Dashboards](#) (2.2.1).

Specifying attributes and relationships for Application Services

Use the Application Service Settings module to specify the attributes and relationships that are required when a user creates an application service.

Configuring the Application Service Settings

Navigate to **CSDM > Application Service Settings** to access the form. For instructions, see [Modify the attributes and relationships required for application services](#).

Application Service Settings

Define required attributes and relationships for application services.

Required attributes	Available	Selected
	Environment Version Model ID Operational Status Support Group Change Group Managed By Group Owned By	> < Number Name

Required relationships	Available	Selected
	Business Application Technical Service Offering Business Service Offering Parent Application Service	> < Cancel Save

Monitoring and managing application services

The Application Service dashboard enables you to monitor and manage application services to ensure that application services are fully configured and are populated in the CMDB.

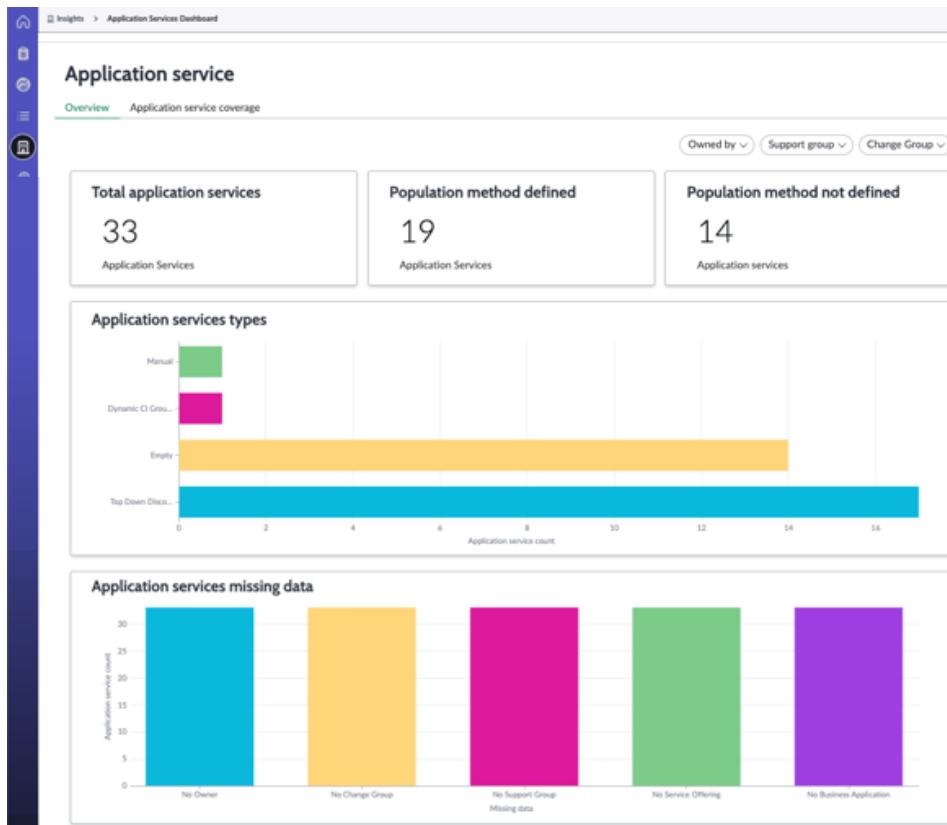
Viewing application service reports

A report on the dashboard, for example, can direct you to an application service that is not configured with a service population method so you can repair it. For more information, see [Monitor the health of application services in the Application Service Dashboard](#).

You have two options:

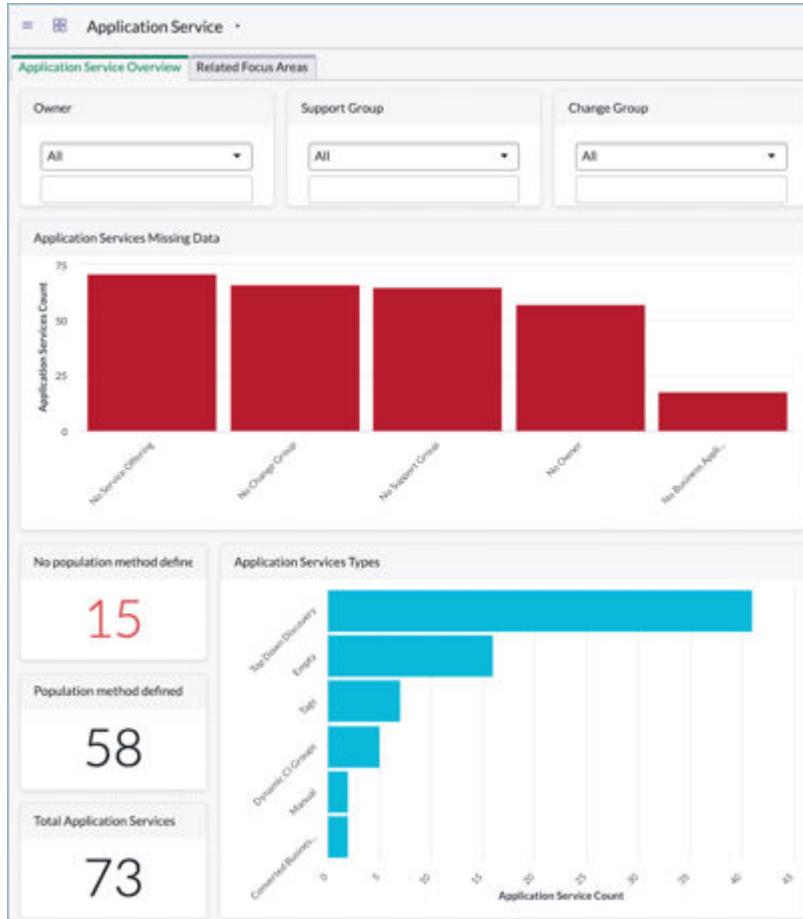
View the Application Service dashboard on CMDB Insights

To view the dashboard in the Insights view, select **CMDB Workspace**, select the Insights icon (💡), and then select the **Application services** tile. See [Insights view in CMDB Workspace](#).



View the Application Service dashboard

To view the dashboard, navigate to **All > CSDM > Application Service Dashboard**.



Synchronizing group assignment attributes

To empower a particular user group to manage a collection of CIs or CI classes, set group assignment attributes through the technical service offering or the CI Class Manager. The operation synchronizes the group attribute data across all CIs that belong to the specified CI class or groups of CIs.

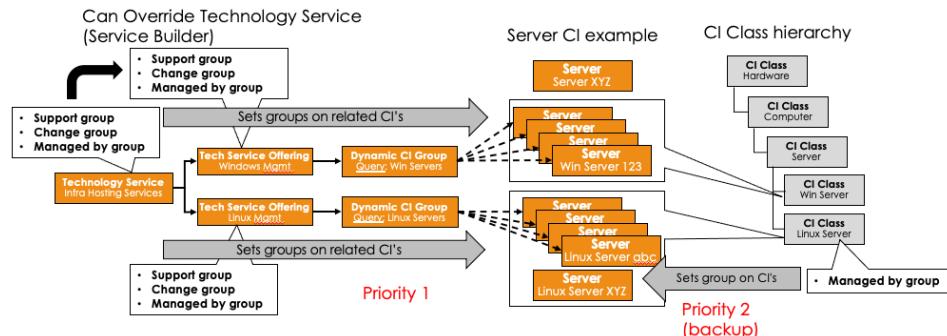
Methods for synchronizing group assignment attributes

- **Synchronize data using a technical service offering:** Directly set the **Support group**, **Change group**, or **Managed by group** attributes in a

technical service offering. The settings are applied to CIs that are associated with the technical service offering.

- Set the group for a CI or an entire class of CIs: Set the **Managed by group** attribute for a specific class in the CI Class Manager. All CIs within the class will have their **Managed by group** field populated based on the value specified in the CI Class Manager. With this method, the **Managed by group** setting is applied only to the CIs that aren't associated with a technical service offering. For CIs that are managed by a technical service offering, the **Managed by group** field is first synchronized with its dynamic CI group. This field is then synchronized with the CIs that are part of that dynamic CI group, overwriting the entry from the CI Class Manager.
- Use Support Group and Change Group: By using dynamic CI groups, data synchronization enables you to manage data that cannot be discovered. The values in the **Support Group** and **Change Group** (previously labeled **Assignment Group**) fields in the cmdb_ci table are synchronized with their related dynamic CI groups and with all the CIs that are contained as part of that dynamic CI group object.

Team data synchronization



- Set the group for a CI or an entire class of CIs

Synchronize group assignment attributes on entire CI classes and individual CIs using the CI Class Manager.

- Synchronize data using a technical service offering

Synchronize group assignment attributes on entire CI classes and individual CIs that use a technical service offering.

Related reference

- [CI Class Manager](#)

Synchronize group assignment attributes on entire CI classes and individual CIs using the CI Class Manager.

Before you begin

Role required: itil and itil_admin

About this task

Set the **Managed by group** attribute for a specific class in the CI Class Manager. All CIs within the class will have their **Managed by group** field populated based on the value specified in the CI Class Manager. With this method, the **Managed by group** setting is applied only to the CIs that aren't associated with a technical service offering. For CIs that are managed by a technical service offering, the **Managed by group** field is first synchronized with its dynamic CI group. This field is then synchronized with the CIs that are part of that dynamic CI group, overwriting the entry from the CI Class Manager.

Procedure

1. Navigate to **All > Configuration > CI Class Manager**.
2. Select **Hierarchy** to expand the CI Classes list, select a class to display details for, and then select **Basic Info**.
3. Specify the value in the **Managed by Group** attribute and select **Save**.
4. To verify the change, select **CI List**.

The change should be applied to the **Managed by Group** attribute of all CIs of this class unless they are associated with a technical service offering. The change is applied only to the CIs in the class and not to the sub classes under a CI.

Example

As an itil_admin, you can define a user group that can manage all CIs belonging to a specific class by following these steps:

1. Navigate to the CI Class Manager and select the **Linux Server** class in the list.
2. Select **Basic Info** and, in the Managed By Group attribute, select **sys_user_group** and then select **Save**.
3. To verify that the attribute was updated, select **CI List** and navigate to the Linux Server class. You will see that the **Managed By Group** attribute has been updated to **sys_user_group**.

Synchronize group assignment attributes on entire CI classes and individual CIs that use a technical service offering.

Before you begin

Role required: itil and itil_admin

About this task

Directly set the **Support group**, **Change group**, or **Managed by group** attributes in a technical service offering. The settings are applied to CIs that are associated with the technical service offering.

Procedure

1. Navigate to **All > Configuration > CMDB Groups** and create a new CMDB group.
See [CMDB groups](#) for details.
2. Navigate to **All > Configuration > Dynamic CI Groups**.
3. Create a new Dynamic CI group and associate it with the CMDB group that you created.

Dynamic CI Group
New record
[New dynamic CI group view*]

Name: dcg1 Support group:

Business criticality: 4 - not critical Change Group:

Owned by: Managed By Group:

Email:

Business phone:

Operational status: Non-Operational

Comments:

Select the CMDB group to create a service based on it

CMDB Group: cmdb-group1

Submit View CMDB Group CIs

See [Manage Technical Services domain of the CSDM framework](#) for more information on Dynamic CI groups.

4. Navigate to **All > CSDM > Technical Service Offering** and create a new technical service offering.
See [Manage Technical Services domain of the CSDM framework](#) for more information on technical service offerings.
5. Navigate to the **CI Relationships** table and select **New** and enter the following values:
 - **Parent:** Select the technical service offering you created.
 - **Child:** Select the Dynamic CI group you created.

The screenshot shows a 'CI Relationship' form titled 'New record'. It has three main input fields: 'Parent' with value 'tool', 'Type' with value 'Contains::Contained by', and 'Child' with value 'dcg1'. Each field has a search icon and a help icon. To the right of the fields is a 'Port' input field. At the bottom left is a 'Submit' button.

6. Select **Submit**.

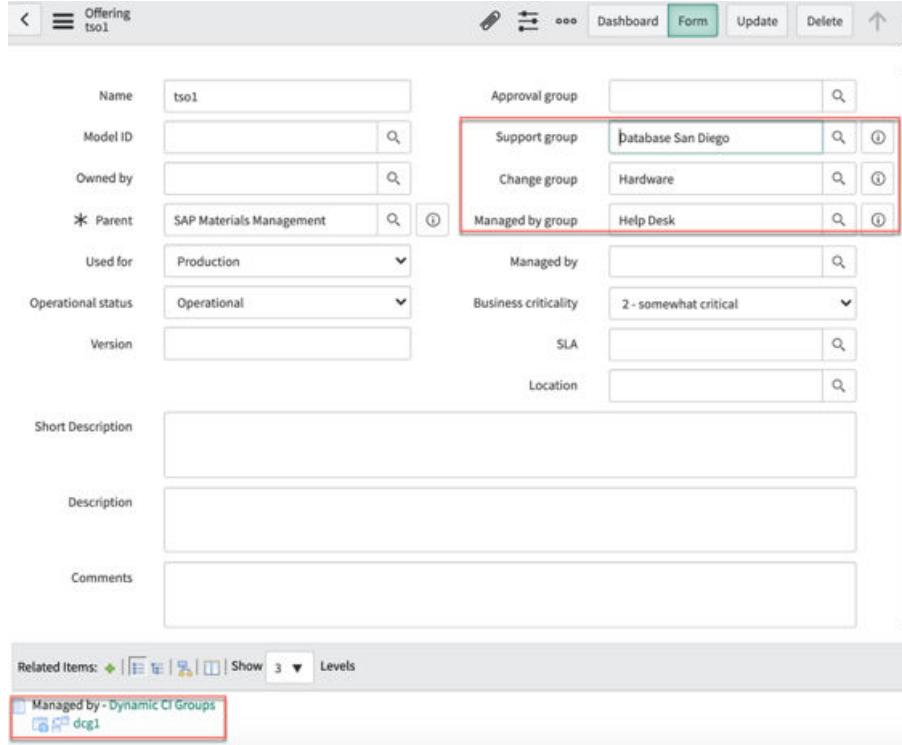
You have created a relationship between the technical service offering and the Dynamic CI group.

7. Navigate to **All > CSDM > Technical Service Offering** and open the technical service offering that you created.

8. Enter values in the following fields:

- Support group
- Change group
- Managed by group

Note: The dynamic CI group that you associated with the technical service offering is listed in the **Managed by** related item.



The screenshot shows the 'Offering' form for item 'ts01'. The 'Managed by group' field is highlighted with a red box. Below the form, the 'Related Items' section shows a link to 'Managed by - Dynamic CI Groups' with the ID 'dcg1', also highlighted with a red box.

Name	ts01	Approval group	
Model ID		Support group	Database San Diego
Owned by		Change group	Hardware
* Parent	SAP Materials Management	Managed by group	Help Desk
Used for	Production	Managed by	
Operational status	Operational	Business criticality	2 - somewhat critical
Version		SLA	
Short Description			
Description			
Comments			

Related Items: Show 3 Levels
[Managed by - Dynamic CI Groups](#)
 dcg1

9. Right-click the header and select **Save**.
A message confirms that data synchronization has been enabled for the fields.
10. The values that you specified are applied to the related Dynamic CI group and all associated CIs.

Note:

- If a new CI is added to the class, the data will be synchronized only after the scheduled CSDM Data Sync job is completed. If you need to synchronize the data immediately, navigate to **Scheduled Jobs > CSDM Data Sync** and select **Execute**.
- The **Managed by Group** field might be populated in both the CI Class Manager and the technical service offering. In this case, the value specified in the **Managed by Group** field in the technical service offering takes precedence.

Applying the CSDM guidelines to your product

The "Product view" topics describe how several ServiceNow products benefit from your use of the Common Service Data Model (CSDM) framework.

Product view topics include the following information:

- Brief overview of the function and benefits of the product.
- Use case and examples.
- CSDM tables managed by the product.
- CSDM tables used by the product.
- Mention of other ServiceNow products that add value to the product.
- Mention of other ServiceNow products that benefit from the product.

Note: A CSDM use case might have one or more data types. Other ServiceNow products may also manage these data types. The use cases don't describe all possible product dependencies.

The product view topics do not address the following information:

- Implementing the ServiceNow product.

- Configuring systems that are not included in the base system. Such systems might be referred to in this product view, but they are out-of-scope.
- Using other ServiceNow products, such as IT Service Management (ITSM), IT Business Management (ITBM), and IT Operations Management (ITOM).

CSDM videos in the ServiceNow Community

[CSDM 4.0 What's New](#)

View the full list of CSDM and data foundation videos.

- [Application Portfolio Management product view](#)

Use APM to gain a comprehensive understanding of your organization's applications so you can identify redundancies and decrease budgetary costs. The goal of this product view is to help you to understand how APM key entities work with the core CSDM framework.

- [Business Continuity Management product view](#)

Business Continuity Management (BCM) gives your organization the capability to continue to deliver products and services at an acceptable level following a disruptive incident. The goal of this product view is to help you to understand how BCM key entities work with the core CSDM framework.

- [Change Management product view](#)

Change Management lets you control every aspect of the IT change process from creation to approval. When you have accurate information, you can minimize risks to your business and avoid conflicts with scheduling. The goal of this product view is to help you to understand how Change Management key entities work with the core CSDM framework.

- [Customer Service Management product view](#)

The CSM enables you to provide service and support for your external customers through communication channels such as the web, email, chat, telephone, and social media. The goal of this use case is to understand how CSM key entities work with the core CSDM framework.

- [DevOps Config product view](#)

The DevOps Config application validates and manages the configuration data of your enterprise applications across every stage of the DevOps pipeline. This ensures that risky changes to configuration data do not get deployed to your production environment. The goal of this product view is to help you to understand how DevOps Config key entities work with the core CSDM framework.

- [Incident Management product view](#)

Incident Management supports the incident management process with the ability to identify and log incidents, classify and prioritize incidents, assign incidents to appropriate users or groups, escalate, resolve, and report incidents. The goal of this product view is to help you to understand how Incident Management key entities work with the core CSDM framework.

- [Operational Technology product view](#)

Operational Technology covers products that tackle aspects of managing OT assets and production processes at various stages of the life cycle. The goal of this product view is to help you to understand how Operational Technology key entities work with the core CSDM framework.

- [ITOM Health product view](#)

ITOM Health includes the Event Management and ITOM Health applications, which help you track and maintain the health of services in your organization. The goal of this product view is to help you to understand how ITOM Health key entities work with the core CSDM framework.

- [ITOM Visibility product view](#)

ITOM Visibility consists of two ServiceNow products: Discovery and Service Mapping. These products are responsible for creating Configuration Items (CIs) in the CMDB and relating them. The goal of this product view is to help you to understand how ITOM Visibility works with the core CSDM framework.

- [Problem Management product view](#)

Problem Management helps identify the cause of an error in the IT infrastructure, reported as occurrences of related incidents. The goal of this product view is to help you to understand how Problem Management key entities work with the core CSDM framework.

- [Service Catalog product view](#)

The Service Catalog lets you create other catalogs (such as the Request Catalog and the Product Catalog) that provide self-service opportunities in the channel you want to use, such as the self-service portal, the mobile app, or the Virtual Agent (conventional interface). The goal of this product view is to help you to understand how Service Catalog key entities work with the core CSDM framework.

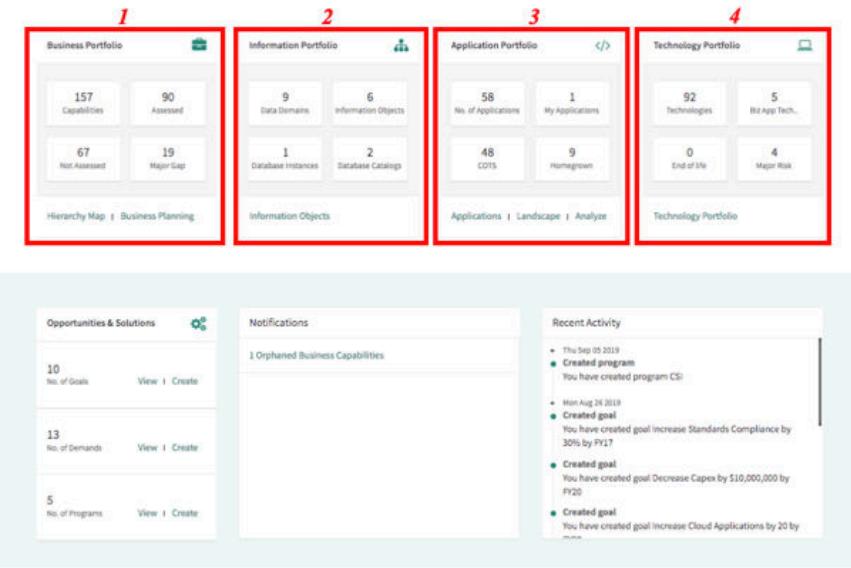
Application Portfolio Management product view

Use APM to gain a comprehensive understanding of your organization's applications so you can identify redundancies and decrease budgetary costs. The goal of this product view is to help you to understand how APM key entities work with the core CSDM framework.

Note: If you do not use CSDM 4.0 or later versions, see the Rome documentation for CSDM.

APM home page

The APM home page organizes many of the CSDM tables used by APM.



Business Portfolio

View the number of defined business capabilities that have been or will be assessed, and the number of business applications that support capabilities but are at-risk.

For more information about using capability mapping to establish a configuration item (CI) relationship between the business capability and the business applications, see [product].

Information Portfolio

Capture the asset information as information objects. You can connect the information object to your business applications to create an application portfolio that you can use at any time.

The information portfolio links to the following data:

- **Data Domains:** Total number of records in the Data Domain table [sn_apm_data_domain].
- **Information Objects:** Total number of records in the Information Object table [cmdb_ci_information_object].

- Database Instances: Total number of records in the Database Instance table [cmdb_ci_db_instance].
- Database Catalogs: Total number of records in the Database Catalog table [cmdb_ci_db_catalog].

For more information about the information portfolio and the information portfolio model, see [product].

Application Portfolio

Track the applications that support your business capabilities and effectively manage them to meet the goals of your organization. The portfolio provides a list of applications with information such as their category, manufacturer, and type. Select **Applications** to navigate to the list view of business applications in your organization.

For more information about measuring the usability, cost, quality, performance, and risk of applications, see [product].

Technology Portfolio

Use metrics to measure the usability, cost, quality, performance, and risk of applications.

For more information about technology portfolio management and how it relates to business applications, see [product].

- [Application Portfolio Management and CSDM tables](#)

Application Portfolio Management manages and uses CSDM tables. Several ServiceNow products benefit from and add value to APM.

- [Application Portfolio Management use case](#)

APM lets you define a single, version-agnostic entity that represents all instances, technologies, and data used for planning and reporting.

- [Application Portfolio Management considerations](#)

Consider these points while implementing the CSDM framework.

Application Portfolio Management manages and uses CSDM tables. Several ServiceNow products benefit from and add value to APM.

CSDM tables managed by APM

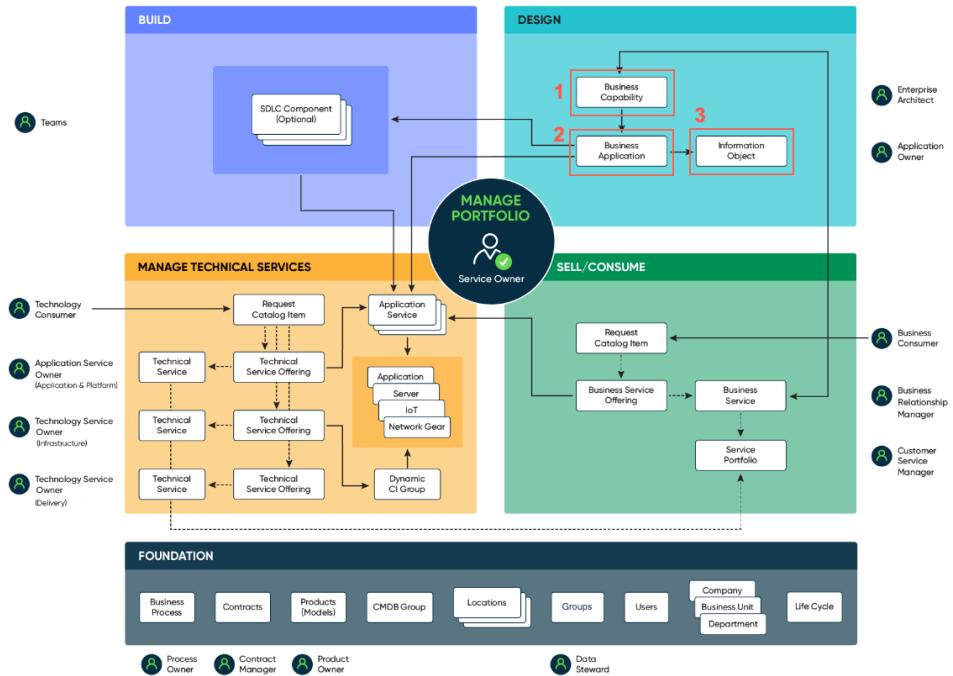
1. Business Capability table [cmdb_ci_business_capability]
2. Business Application table [cmdb_ci_business_app].

Note:

APM uses the **Platform** and **Platform App** fields on the Business Application table to establish the relationship between a business application and the underlying application service. APM manages the Platform Host / Platform App relationship using a reference on the APM form, not through a CI relationship.

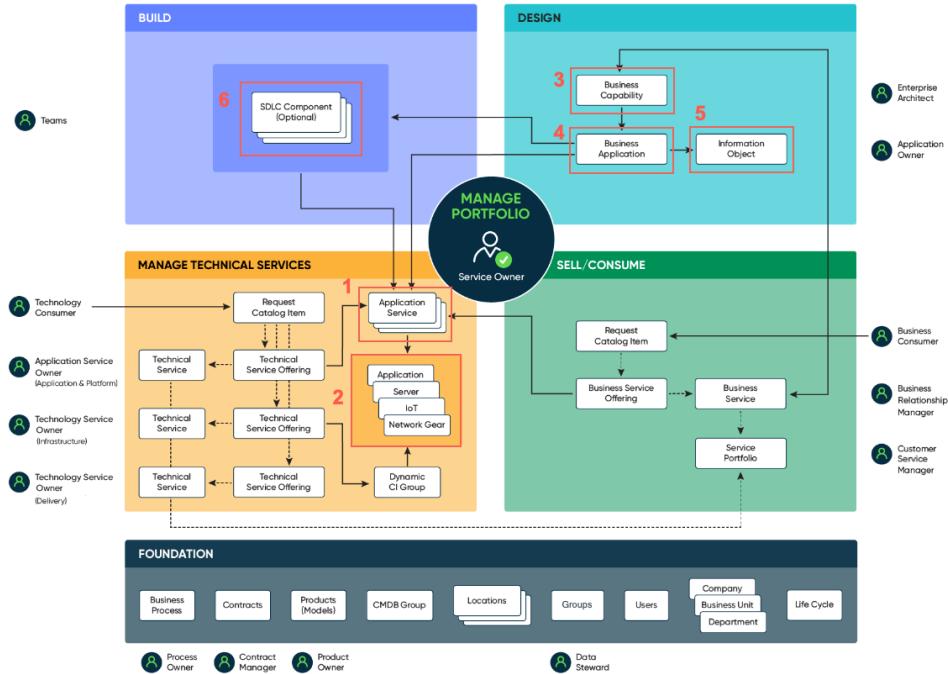
The relationship connects the record of the business application that is used in planning and design with where and how it's realized operationally, represented by application services. The relationship accounts for each use of a business application in the development, test, and production environments (dev, test, and prod application service instances). Often there are multiple production deployments. For example, a large retailer uses a business application that runs a cash register in each of its 1,000 stores. There are therefore 1,000 production instances of the application service — one per store — for that one business application. [See the "CSDM in a nutshell" video](#) for additional discussion of the relationship.

3. Information Object table [cmdb_ci_information_object]



CSDM tables used by APM

1. Mapped Application Service table [cmdb_ci_service_discovered]
2. Configuration Item tables [cmdb_ci*]
3. Business Capability table [cmdb_ci_business_capability]
4. Business Application table [cmdb_ci_business_app]
5. Information object table [cmdb_ci_information_object]
6. SDLC Component table [cmdb_ci_sdlc_component]



Products that add value to APM

When you use APM with any of the following ServiceNow products, you increase the value you get from APM:

- Discovery provides details about the hardware and software CIs you are using.
- Service Mapping provides details about the application instance service in the [cmdb_ci_service_discovered] table, relating infrastructure and application [cmdb_ci_appl] CIs.
- Asset Management provides the related product model. Software Asset Management (SAM Foundation) and Hardware Asset Management (HAM) provide life-cycle data for Technology Portfolio Management.
- Financial Management: Total Cost of Ownership (TCO) calculation based on general ledger data and allocation rules.
- Project Portfolio Management views the business application roadmaps. Includes demands, projects, sprints and epics.

- Agile Development views the backlog stories and epics of each business application in the application roadmap.

Products that benefit from APM

The following ServiceNow products gain value from APM:

- IT Service Management (ITSM): Services have the context of the business and applications, along with the information and technologies layered beneath them.
- Information Technology Operations Management (ITOM): Understands the business context for the application services along with the hardware and software being managed.
- Governance, Risk, and Compliance (GRC): Auditors can leverage the business applications and related information objects. This helps auditors understand the design-time data sensitivity for scoping audits, measuring risks, and managing audit activities.
- Asset Management: Manages the software and hardware life cycles for business applications and business services.

APM lets you define a single, version-agnostic entity that represents all instances, technologies, and data used for planning and reporting.

APM use case

You can use a business application for planning and governance activities, such as funding, road mapping, and risk reporting. Rationalizing business applications is a continuous process, and is critically important to reducing costs and planning technology transformations. Rationalizing business applications is also critical for completing mergers, divestitures, or other broad-impact business-led changes.

Key features of the APM use case

The CMDB, when used by the CSDM framework, provides value to APM in the following ways:

- Application life cycle management. This includes:
 - Registering a new business application (included in the base system).
 - Updating a business application

- Decommissioning a business application, including all the related application services and infrastructure. Because application services are logical in nature, they should use the Logical life cycle states. Application services follow the same life cycle guidance as any other logical CI.
- Business application portfolio assessments based on metrics or related impacts.
- Roadmap planning and creating new ideas, demands and projects.
- Data certification process
- Total cost of ownership (TCO) calculations (using the Financial Management module)
- Manage the following related entities:
 - Information objects table [cmdb_ci_information_object]
 - Business capabilities table [cmdb_ci_business_capability]

Results of the APM use case

With this use case, CSDM provides APM a consistent way to model business applications and relate critical data. The use case ensures that the application services (instances) are defined as required for automating the technology risk scores, costs, and other metrics used for analysis.

TPM use case

TPM gives you a better understanding of the risks associated with using software and hardware that is at the end-of-life (EOL) date. You can use the details provided by the CSDM framework to determine the risk of using software and hardware that is at EOL. Each product life cycle EOL date is calculated, then combined following the CSDM framework to provide a score at the Business Application level.

Results of the TPM use case

The CSDM framework provides a consistent data structure. This consistent data structure makes it easier for you to manage the life cycles of your technology and analyze the combined technology risks.

Because of the way the CSDM framework is structured, you can leverage many products from ITOM, Service Management (Service Portfolio Management), and IT Application Management (ITAM).

The risks of using EOL technologies are calculated based on the life cycle of each software and hardware product model identified in the CMDB, and matched with a software and hardware product model.

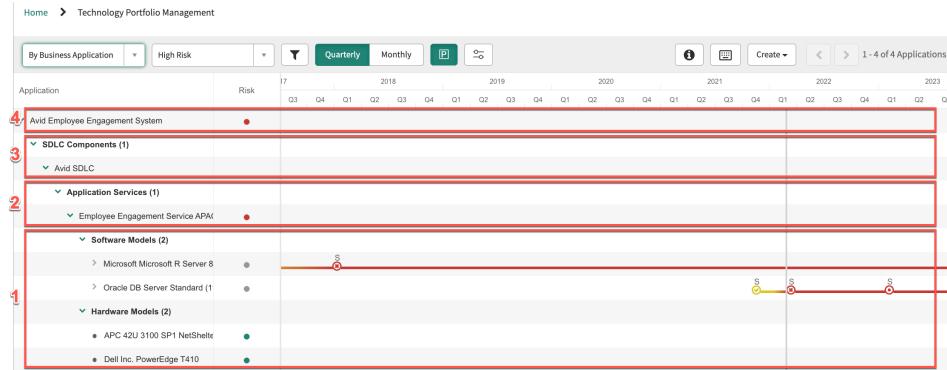
You can enter the life-cycle data manually, import it from an external source, or use the data provided with your Software Asset Management Professional or Hardware Asset Management license.

The risks are calculated and displayed in a hierarchy. Business application is at the top level, SDLC Components are under the business application, then Application Services indicate each deployment (instance), and software and product models are at the lowest level. Risks are calculated in the order shown below, and are based on the time span between the current date and the EOL date.

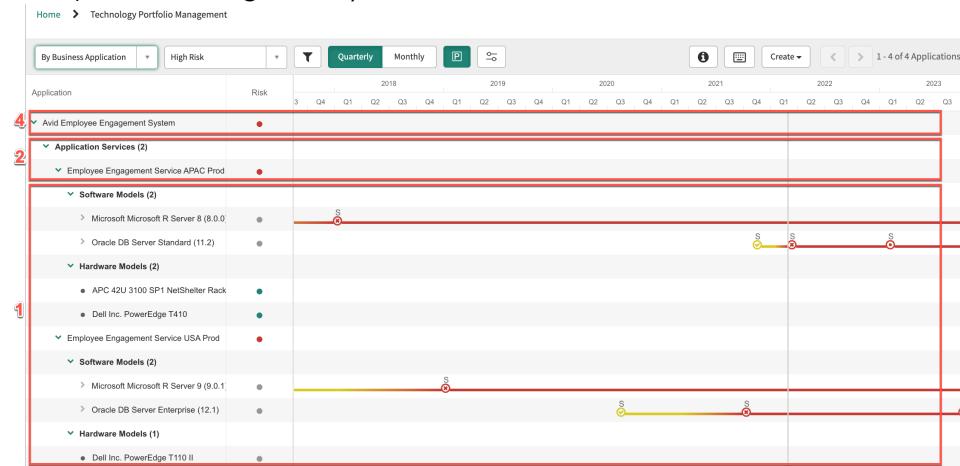
Note: Configuring SDLC component is optional. Even without SDLC component configuration, you can connect business applications with the application services directly.

1. Hardware and software product model — Displays the current life-cycle phases, sources, and indicates the specific models at-risk
2. Application Service level — Displays the combined risk status of all underlying hardware and software product models used in the Application Service (Instance).
3. SDLC Component — Displays the SDLC components along with the associated application services and business applications
4. Business application level — Combines all the underlying Application Service (Instances) to determine the overall risk rating at a portfolio level.

Technology Portfolio Management home page (with the SDLC Component configuration)



Technology Portfolio Management home page (without SDLC Component configuration)



The following information is used to determine the EOL impact to business applications and their installed application services (instances):

- The business applications used in your organization are all linked to one or more application services. Each of the application services run on one or more technologies or software models.

The name of the Application Service Software model table is [sn_apm_tpm_service_software_model].

- The software model has a sequence of life-cycle stages. The life-cycle stages range from the installation date to the retirement date.

Some business organizations set an internal date based on the life-cycle phase of the software models. These software model phases can be Early Adopter, Mainstream, Declining use, and Retired.

Similarly, the software vendors might also set a date for the software based on the vendor life-cycle phases, such as Pre-release, General Availability, End of Life, and Obsolete. Vendor support might vary depending on the phase of the technology. For example, when the software model reaches the Obsolete phase, the vendor might stop supporting the technology.

The Software Model Life cycle table is named [sam_sw_model_lifecycle].

Consider these points while implementing the CSDM framework.

-

Business services are what IT provides to the customer. A business service is a service type that is published to business users. A business service typically implements one or more business capabilities.

Usually, business users order business services. Business users can select the desired offering and service commitment levels via the Service Catalog. For example, procurement, shipping, and finance.

- A business service is an operational CI.
- A business service must be a one-level service and not a hierarchy of business services.
- A business service can be used for impact in Incident, Problem, and Change and for approvals for Change.
- A business service must be focused on the consumer or seller.

-

A business capability is a high-level capability that supports a business model or fulfills a mission for your organization.

A business capability typically describes a specific task that achieves one or more business outcomes. Business capabilities are often listed as verbs (for example, manage financials or provide IT support services).

You can use business capabilities to rationalize and prioritize the cost of business applications and business services.

- Using Technology Portfolio Management: The Software Asset Management Foundation plugin provides life-cycle data that Technology Portfolio Management uses. However, you can manage that data manually or get it from another source. Both products share the underlying tables, but are independent and you can use them separately.
- For information about managing your application portfolio, see [Application Portfolio Management - Inventory Best Practices](#).
- Applications and business applications use different tables and represent different elements.

Business application

- Uses the Business Application table [cmdb_ci_Business_App].
- Represents the single, logical, construct of the application that comprises application service, environment, software, and hardware in use.

Application

- Uses the Application table [cmdb_ci_appl].
- Represents the specific version of software in use on a server (often populated by Discovery or System Center Configuration Manager (SCCM)).

For additional information about APM, see [APM: Application Inventory - Most common questions](#).

Business Continuity Management product view

Business Continuity Management (BCM) gives your organization the capability to continue to deliver products and services at an acceptable level following a disruptive incident. The goal of this product view is to help you to understand how BCM key entities work with the core CSDM framework.

The business impact analysis capability enables business teams to identify the IT and Non IT dependencies that can be captured as part of the CMDB.

The same information is used to create IT DR and BC plans along with the recovery steps.

Use BCM to identify the objective, governance model, and framework for effective implementation and maintenance of a BCM program. The functional components of BCM operate within the framework, interact with each other to respond to a crisis situation in a synergistic manner, and execute end-to-end critical processes to minimize impacts on employees, business, and customers.

BCM includes the following major functional components to alleviate the disruption to your organization, continue operations, and deliver your business services during a disruption.

- Business Impact Analysis
- Business Continuity Planning and Recovery Management
- Crisis Management

For more information

For more information about using Business Continuity Management see:

- [Business Impact Analysis](#)
- [Business Continuity Planning / IT DR planning](#)
- [Plan Exercising](#)
- [Crisis Events](#)
- [Business Continuity Management and CSDM tables](#)

Business Continuity Management manages and uses CSDM tables. Several ServiceNow products benefit from and add value to Business Continuity Management.

- [Business Continuity Management use case](#)

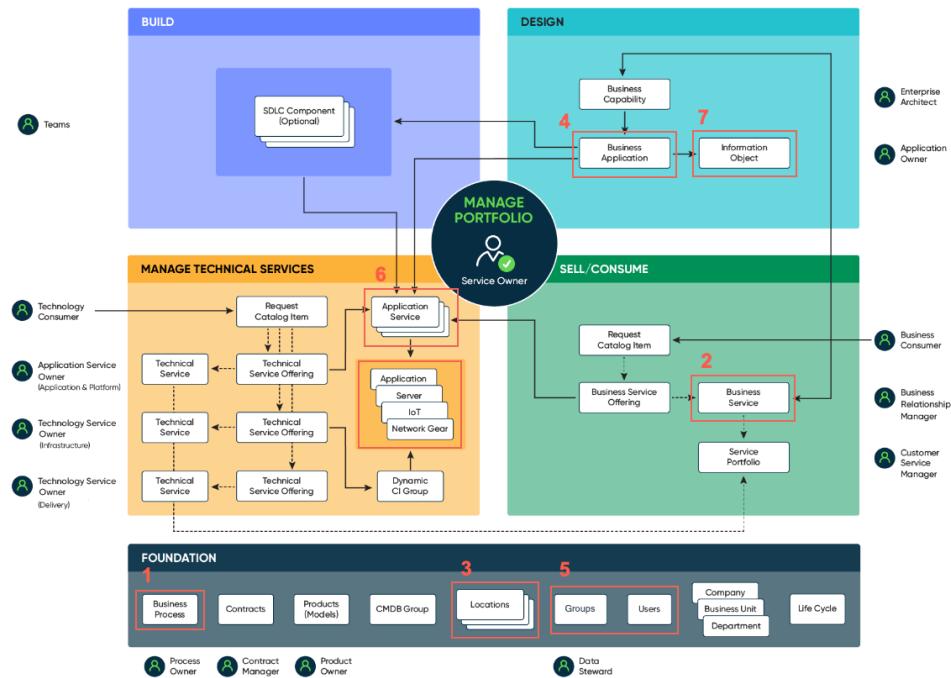
Business Continuity Management use cases are described in this section.

- [Business Continuity Management considerations](#)

Consider these points while implementing the CSDM framework.

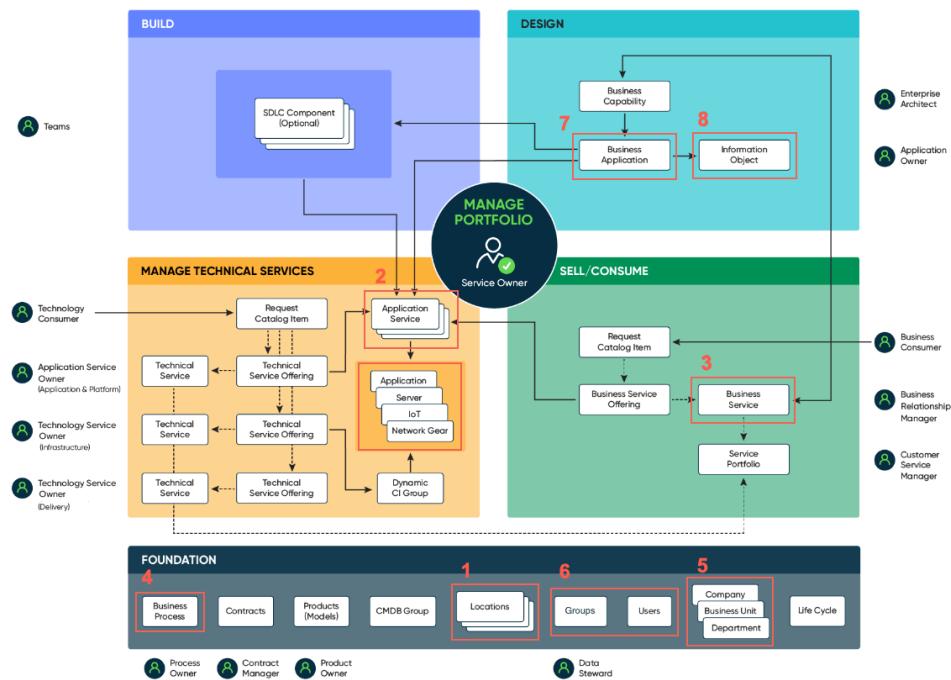
Business Continuity Management manages and uses CSDM tables. Several ServiceNow products benefit from and add value to Business Continuity Management.

CSDM tables managed by BCM



1. Business Process
2. Business Service
3. Locations
4. Business Applications
5. Users and Groups
6. Application Service
7. Information Object

CSDM tables used by BCM



1. Locations table [cmn_location]
2. Mapped Application Service table [cmdb_ci_service_discovered]
3. Configuration Item table [cmdb_ci*]
4. Business Process table [cmdb_ci_business_process]
5. Business Unit and Department tables [business_unit, cmn_department]
6. Users and Groups user table [sys_user]
7. Business applications table [cmd_ci_business_appl]
8. Information Object table

Products that add value to BCM

When you use BCM with other ServiceNow products, you increase the value you get from Business Continuity Management.

Integrated Risk Management (IRM), for example, provides a holistic approach to the BCM program. The risk assessments become a critical input for creating the business continuity planning phase. The BCM policy can be authored and maintained in the policy and compliance capability of IRM.

Products that benefit from BCM

Integrated Risk Management

Risk assessments can be scoped based on the business impact analysis done on business services, business processes and applications.

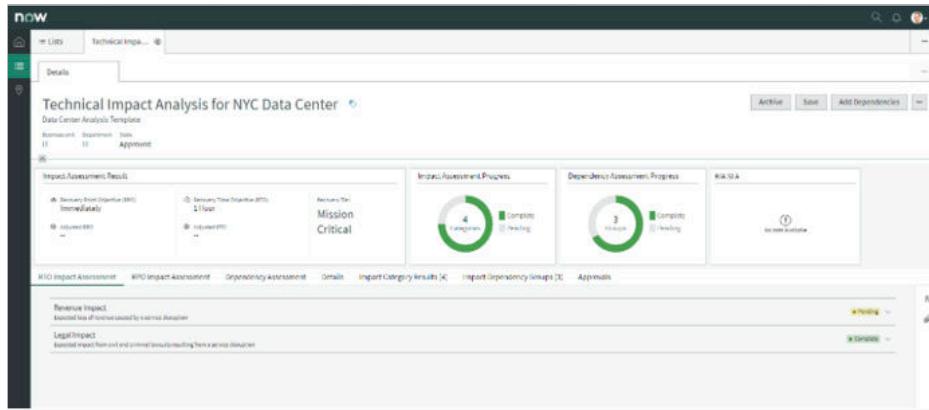
Operational Resilience

The business continuity plan status and exercise results are used to track the resilience profile of the business services.

Business Continuity Management use cases are described in this section.

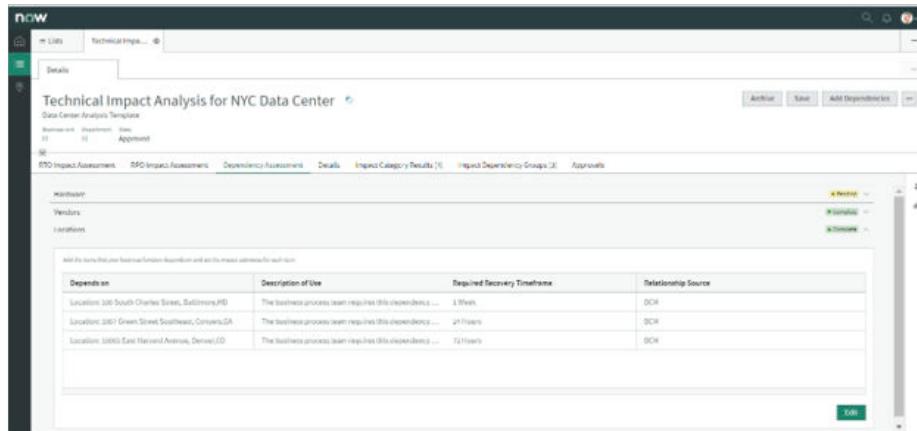
Business Impact Analysis use case

The Business Continuity Managers are enabled with BIAs. In the BIAs the users can analyze the financial and non-financial impact of a service, process, CI downtime to classify the CMDB data into recovery tiers, and identify the recovery objectives.



The managers can also identify the CI and non-CI dependencies like:

- CIs: Business applications, application services, servers, hardware.
- Non-CIs: Business services, business processes, locations, users, groups, business units, departments, suppliers.



Business Continuity Planning and Recovery Management use case

Business continuity managers can create and manage business continuity and IT Disaster Recovery (DR) plans by referencing the CIs.

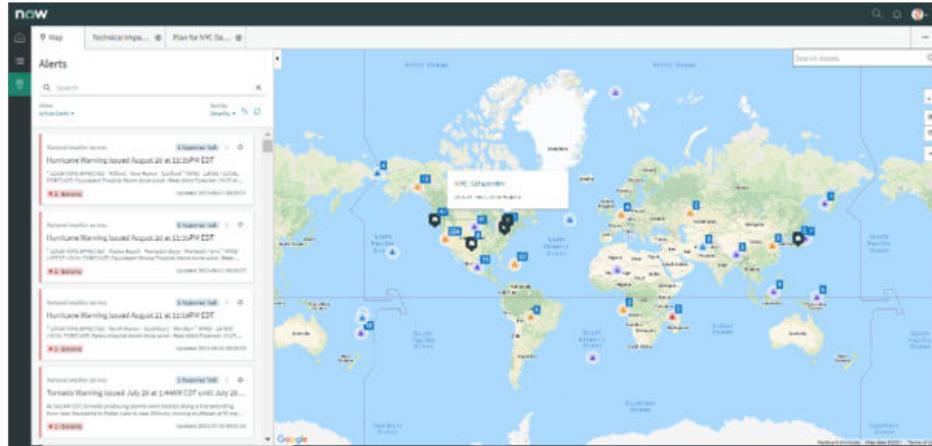
Name	Recovery Time Objective	Recovery Point Objective	Note
Data Center: NYC Datacenter	0 Hours	1 Day	Impact Analysis: Technical Impact Analysis for NYC Data Center

Business continuity managers can identify recovery strategies and instructions to recover the CIs and non-CIs in case of a disruptive event.

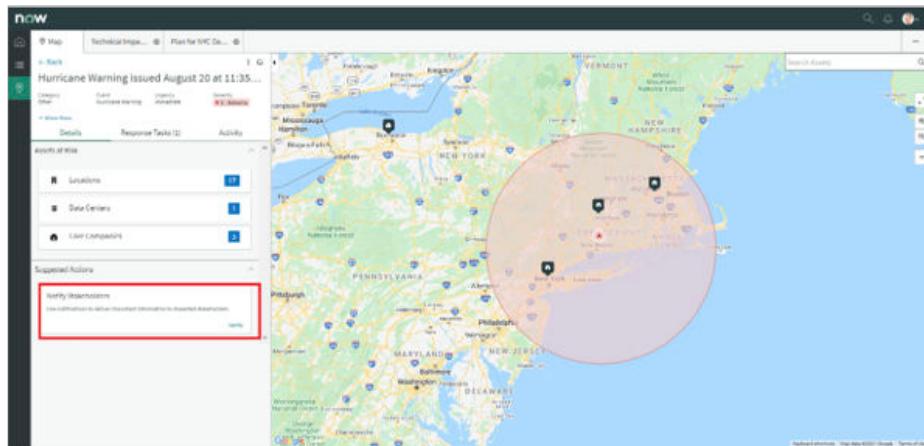
Name	Comments	Dependencies Covered	Description	Maximum Duration of Use	Estimated % of Operations Achieved	Estimated Time to Implement
Alternative work location instructions		Location: 100 South Charles Street, Baltimore, MD		2 Weeks	90	4 Hours
Hand over operations to offshore team		Location: 10389 Democracy Lane, Fairfax, VA		1 Week	90	4 Hours

Crisis Management use case

Crisis managers can identify the impacted CIs and non-CI items like datacenters, locations, business units due to a crisis event.



Crisis managers can alert the stakeholders of items under risk, send notifications, and activate recovery plans related to the CI and non-CI information.



Consider these points while implementing the CSDM framework.

- Difference between a Business Capability and a Business Service:

A business service is a service type that is published to business users. A business service typically implements one or more business capabilities.

Usually, business users order business services. Business users can select the desired offering and service commitment levels via the Service Catalog. For example, procurement, shipping, and finance.

- A business service is an operational CI.
- A business service must be a one-level service and not a hierarchy of business services.
- A business service can be used for impact in Incident, Problem, and Change and for approvals for Change.
- A business service must be focused on the consumer or seller.

A business capability is a high-level capability that supports a business model or fulfills a mission for your organization.

A business capability typically describes a specific task that achieves one or more business outcomes. Business capabilities are often listed as verbs (for example, manage financials or provide IT support services). You can use business capabilities to rationalize and prioritize the cost of business applications and business services.

- Using Technology Portfolio Management: The Software Asset Management Pro provides life-cycle data that Technology Portfolio Management uses but that data can be managed manually or can be loaded from alternate source. The underlying table structures are shared between both products but both products run independent of one another.
- Managing an Application Portfolio: See [Application Portfolio Management - Inventory Best Practices](#) for details on how to manage a business application.
- Difference between a business application and an application:
 - A business application (cmdb_ci_business_app) represents the single, logical construct of the application that is made of all application services including the environment, software, and hardware that has been deployed.
 - An application (cmdb_ci_appl) represents the specific running installation of software running on a specific server, often populated from discovery or SCCM.

Change Management product view

Change Management lets you control every aspect of the IT change process from creation to approval. When you have accurate information, you can minimize risks to your business and avoid conflicts with scheduling. The goal of this product view is to help you to understand how Change Management key entities work with the core CSDM framework.

Change Management

Change Management includes the following features:

- Manage changes more quickly by using the Change Advisory Board (CAB) Workbench to schedule, plan, and manage CAB meetings from one interface.
- The Change Management backlog analysis dashboard provides increased visibility into any changes.
- Service Maps let you see the change impacts at-a-glance.
- The change approval policies increase DevOps velocity and remove IT friction.

Change Request form

The Change Request form references the following attributes and related lists.

1 Service (Business Service)	References the [cmdb_ci_service_business] table. Note: Earlier platform releases labeled this attribute Business Service .
2 Service Offering	References the [service_offering] table where the offering has a parent service.

3 Configuration Items	References the [cmdb_ci] table.
4 Affected/Causal Cls	Related list [task_ci] table.
5 Impacted Services	Related list [task_cmdb_ci_service] table.
6 Assignment Group	<p>References the Group attribute.</p> <p>Note: You can populate the Group attribute by using the Assignment Group for the relevant CI.</p>

The screenshot shows the Change Request (CR) form with various fields and sections highlighted with red boxes:

- 1 Business service**: A dropdown menu under Category.
- 2 Service offering**: A dropdown menu under Category.
- 3 Configuration item**: A dropdown menu under Category, set to "OWA-SO-01".
- 4**: A section for conflict status and last run time.
- 5**: A section for assignment group and assignee.
- 6**: A section for planning details like start and end dates.

Below the main form, there is a related links section with "Affected CIs" and "Impacted Services/CIs" highlighted with red boxes, and a search bar at the bottom.

For more information

For additional details on Change Management, see [Change Management](#).

See the video: How Change Management leverages the CSDM

- [Change Management and CSDM tables](#)

Change Management manages and uses CSDM tables. Several ServiceNow products benefit from and add value to Change Management.

- [Change Management use case](#)

For ITSM, specifically incident and change, identifying the location of critical data can help reduce mean time to resolve incidents and eliminate outages caused by change.

- [Change Management considerations](#)

Consider these points while implementing the CSDM framework.

Change Management manages and uses CSDM tables. Several ServiceNow products benefit from and add value to Change Management.

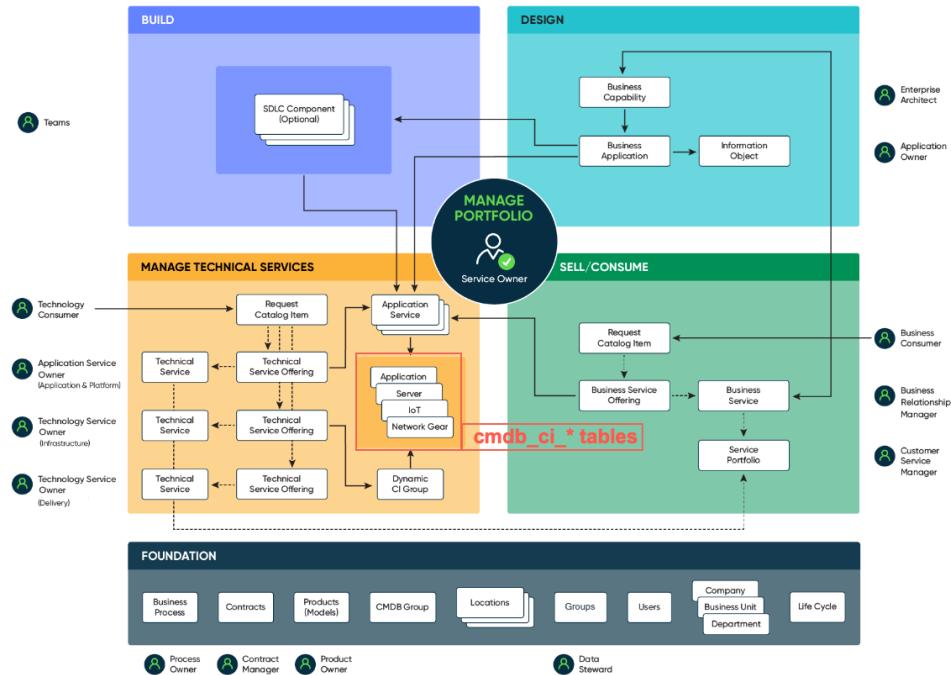
CSDM tables managed by Change Management

Configuration items (CIs) tables [cmdb_ci_*]:

- Application table [cmdb_ci_appl]
- Server table [cmdb_ci_server]
- Virtual machines table [cmdb_ci_vm_instance]
- Load balancer table [cmdb_ci_lb]
- Network gear table [cmdb_ci_netgear]

The following figure highlights the CSDM tables managed by Change Management.

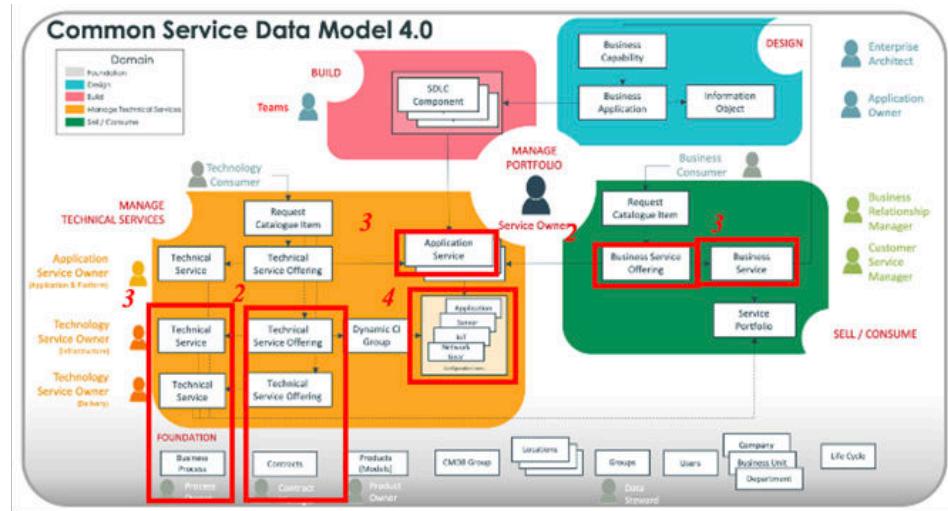
CSDM tables managed by Change Management



CSDM tables used by Change Management

1. Application Service table [cmdb_ci_discovered_service] or any infrastructure CI.
2. Configuration Item tables [cmdb_ci*]
3. Business Service [cmdb_ci_service_business] table and Technical service [cmdb_ci_service_technical] table: Uses the service classification attribute to identify business services, technical services, and application services as types of services.
4. Service Offering [service_offering] table: Uses the service classification attribute to identify business services, technical services, and application services as types of service offerings.

Tables used by Change Management



Products that add value to Change Management

When you use Change Management with other ServiceNow products, you increase the value you get from Change Management. These other ServiceNow products include:

Discovery

Discovery provides details about the hardware and software CIs you are using.

Service Mapping

Service Mapping provides details about the application instance service in the Mapped Application Service [cmdb_ci_service_discovered] table, relating infrastructure and application [cmdb_ci_appl] CIs.

Products that benefit from Change Management

Service Portfolio Management (Service Portfolio Management)

Services have the context of the business and applications, along with the information and technologies that underpin them.

IT Service Management (ITSM)

Incidents caused by, impacted by, or fixed by Change Management.

Asset Management

Updates assets.

Information Technology Operations Management

Updates CIs using a controlled process.

DevOps

Provides Change Management governance for pipeline tool chains.

For more information

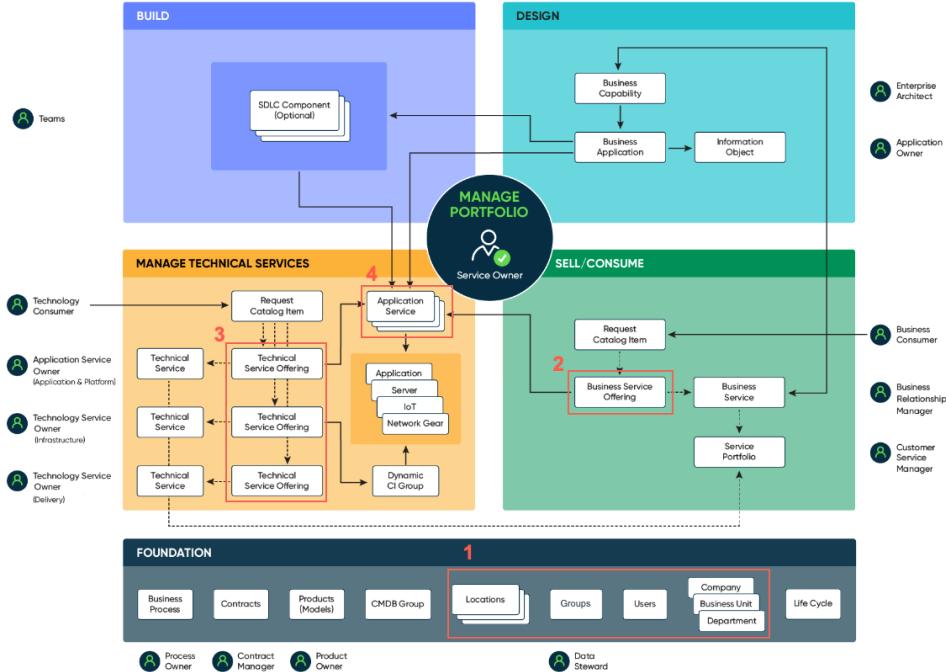
[See the video: How Change Management leverages the CSDM](#)

For ITSM, specifically incident and change, identifying the location of critical data can help reduce mean time to resolve incidents and eliminate outages caused by change.

Key features of the Change Management use case

Applying the CSDM framework provides value to Change Management in the following ways:

- Enables users to understand the impact of a change on services and service offerings.
- Changes are dynamically routed.
- Change Management identifies and notifies all affected services to support the approval decision.



- Subscription: Related lists on service offerings that identify who has access to the offering and thus may be impacted in an outage. An incident or change can identify impact using the subscribed by tables. The related lists are as follows:

- Service Subscriptions by Company [service_subscribe_company]
- Service Subscriptions by Department [service_subscribe_department]
- Service Subscriptions by Group [service_subscribe_sys_user_grp]
- Service Subscriptions by Location [service_subscribe_location]
- Service Subscriptions by User [service_subscribe_sys_user]

- Business service offering may be used to provide the business approver based on approval_group and business_criticality. A business service may have multiple offerings, each with a different criticality.

3. Technical service offering may be used to provide the technical approver approval_group and technical assignment group on the attribute assignment_group. May be used by change for routing of change and change tasks. May be synchronized onto the CI's that the offerings manage thus reducing the manual overhead of maintaining manual data on thousands/millions of CI's.
4. Application service may be used to provide prod and non-prod (DEV, QA, UAT, etc.) environments. Non-prod environments may be filtered out if desired. The legacy **used_for** attribute maps to the **environment** attribute. You should use the **environment** attribute.

Note: Some service offerings may identify the **environment** of the offering as well.

Results of the Change Management use case

The CSDM framework provides context for the changes. The context includes the CIs involved in the change and the services affected.

Use the Change Request form to see the impact of the change.
Complete the following steps:

1. Populate the Configuration Item attribute [configuration_item] with the target CI for the change activity. You can then use this CI to identify details for change routing. For example, you can use the CI data, such as "Assignment Group" or "Approval Group," and provide information about the service impact by using dependency relationships.
2. Populate the Impacted Services related list [task_cmdb_ci_service] with the services that are related to the populated CI. These may include services and service offerings.
3. (Optional) Use the Service and Service Offering attributes to identify the provider services responsible for managing the selected CIs.
4. (Optional) Use the Affected CI related list [task_ci] to identify the CIs that may have caused the change. These CIs are in addition to the CIs previously populated. The [task_ci] table can be populated dynamically or manually.

Note: Dynamic population is not part of the base system. To use dynamic population, you need to configure the Change Request form.

For more information

[See the video: How Change Management leverages the CSDM](#)

Consider these points while implementing the CSDM framework.

- Business applications not referenced in the Change Management use case: Business applications are portfolio objects you can use for designing and planning an Enterprise Architecture. Business application portfolio objects don't contain version, environment, and localization details for deployments using one or more applications.
- Change approval process: It depends on how you implement Change Management. After you populate the CIs on the Change form (if there's a relationship between the CI and the impacted services and service offerings), the Approval group approves the change.
- CI attributes used for routing changes: If you are initiating the change or the change task, use the Assignment Group attribute CI. If you are using build run teams, you could use the Support Group attribute CI for the team assignments.

For more information

[See the video: How Change Management leverages the CSDM](#)

Customer Service Management product view

The CSM enables you to provide service and support for your external customers through communication channels such as the web, email, chat, telephone, and social media. The goal of this use case is to understand how CSM key entities work with the core CSDM framework.

CSM use case

CSM provides proactive customer service, decreases cost of service, and provides end-to-end visibility to both customer service and service

delivery groups. For more information, see [Integrating with Service Portfolio Management](#).

For information on setting up the data, see [Configure foundation data](#).

Results of the CSM use case

Understand the tables within the CSDM framework that are needed to support the following CSM functionality:

- Service-aware Install Base
- Proactive Customer Service Operations
- Real Time Service Health and Outage Tracking
- Contextual Service Catalog

The following sections include details of the activities needed for the use case.

CSM data model

CSM uses the following key entities to resolve complex end-to-end issues.

- A **case** is the primary entity of CSM. CSM uses cases to track and resolve customer questions or issues.
- Customer information is linked to a case using associated entities such as Accounts, Contacts, Consumers, Household, Products, and Service Contracts. This information provides the customer service agent the information necessary to resolve customer issues.
- An **account** can be a customer account, a partner account, or both.
- The **contact** is an employee of an account. A contact record stores information about a contact, such as the name, phone number, and email address. A contact can also have a user ID and can log in to the customer portal.
- A **consumer** is a customer in the business-to-consumer (B2C) business model.
- Household defines the consumers who constitute a household, and the relationships between household members.

- Sold Product tracks the products or services sold to an account or consumer.
- Install Base Items represent the instance of the product that has been configured for a customer.
- Service Contract defines the type of support that customers receive. A contract can include an account and contact or a consumer and the specific assets that are covered. A contract can also include multiple service entitlements and SLAs.
- Entitlement specifies the type of support that a customer receives as well as the supported communication channels.
- [Customer Service Management and CSDM tables](#)

Customer Service Management manages and uses CSDM tables. Several ServiceNow products benefit from and add value to Customer Service Management.

- [Customer Service Management examples](#)

In this example, the CSDM helps the fictitious ACME technology company use CSM to manage network monitoring services for customers.

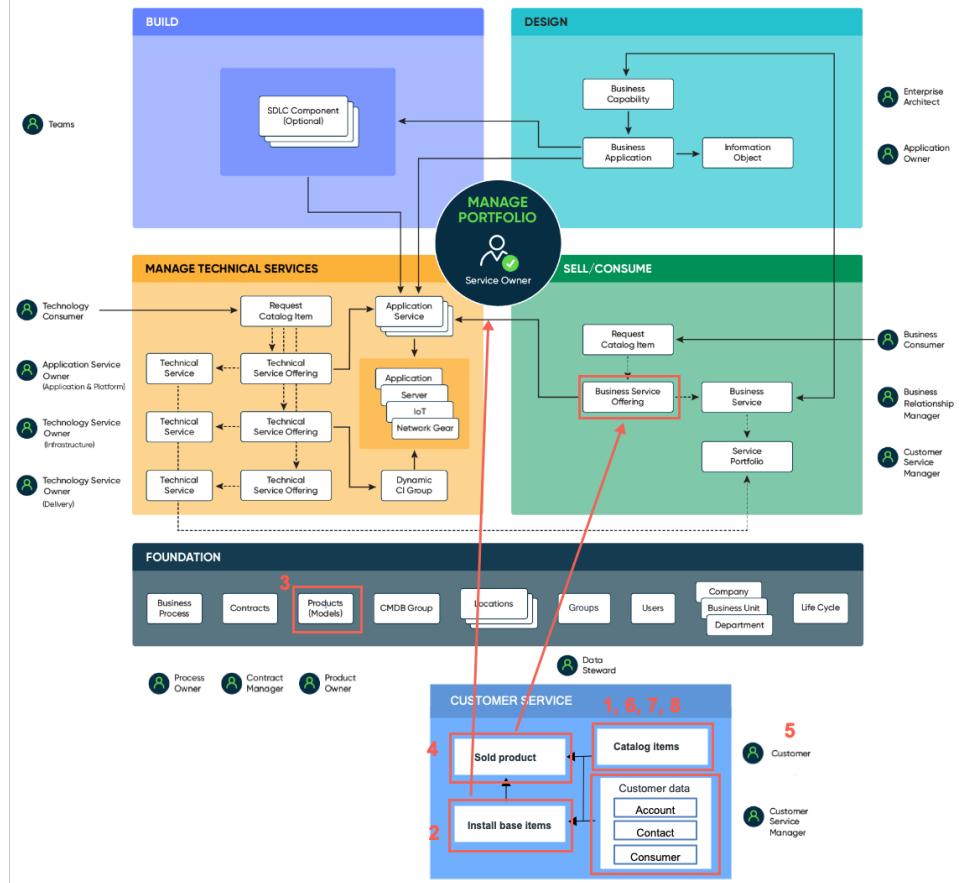
- [Customer Service Management FAQs](#)

Consider these points while implementing the CSDM framework.

Customer Service Management manages and uses CSDM tables. Several ServiceNow products benefit from and add value to Customer Service Management.

CSDM tables managed by CSM

CSM tables managed by CSDM



CSM references the following CSDM tables:

1. Sold Product table [sn_install_base_sold_product] in CSM

Represents the product purchased by an Account or Consumer, and references the Product Model table [cmdb_model] or Service Model table [cmdb_service_product_model] for a Customer (Account or Consumer).

2. Install Base Item table [sn_install_base_item] in CSM

Represents the products installed or in use by an account or consumer. Install Base Items are CIs consumed by the

customer and generally reference the Application Services table [cmdb_ci_discovered_service] for SaaS products.

Multiple sold products can be used on a given Install Base Item by using the Installed Products table [sn_install_base_m2m_installed_product].

3. A Service Model references the Service Offerings table [service_offering]. Multiple Service Offerings can be associated with a single Service Model.
4. After the product is sold to a customer, the Sold Product table references the Service Offering table [service_offering]. This reference helps to identify the customers subscribed to an offering.
5. Customers can request services related to the products they have purchased by linking the Catalog Items table [sc_cat_item] to the Product Model or Service Model by using the Product-to-Catalog Items Relationships table [sn_prod_cat_rel_m2m_product_catalog_item].
6. Account table [customer_account] in CSM

Extends the Company table. The Account table can be a customer account, a partner account, or both.

7. Contact table [customer_contact] in CSM

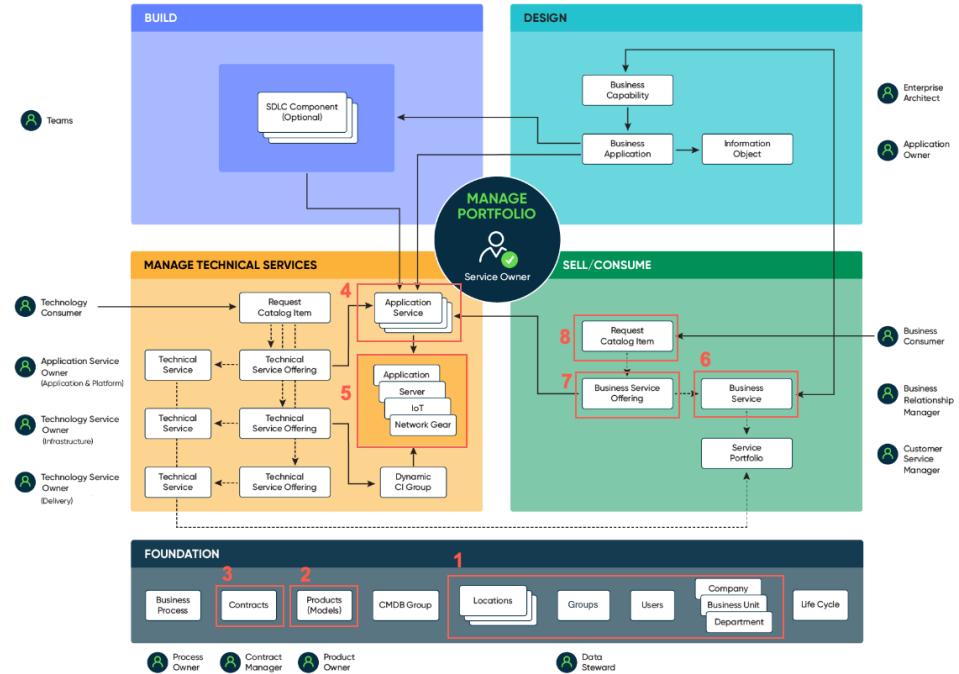
Extends the User table. A user is an employee of an account. A contact record stores information about a contact, such as the name, phone number, and email address. A contact can also have a user ID and can log in to the customer portal.

8. Consumer table [csm_consumer] in CSM

A consumer is a customer in the business-to-consumer (B2C) business model.

CSDM tables used by CSM

CSDM tables used by CSM



1. Company [core_company], Business Unit [business_unit], Department [cmn_department], Location [cmn_location], Groups [sys_user_group], Users [sys_user]
2. Product Model tables [cmdb_model], and [cmdb_service_product_model]
3. Contract table [ast_contract]
4. Mapped Application Service table [cmdb_ci_service_discovered]
5. Configuration Item table [cmdb_ci_*]
6. Business Service table [cmdb_ci_service_business]
7. Business Service Offering table [service_offering]
8. Request Catalog table [sc_cat_item]

Products that add value to CSM

When you use CSM with other ServiceNow products, you increase the value you get from CSM. These other ServiceNow products include:

IT Service Management (ITSM)

Services have the context of the business, applications, information, and technologies layered beneath them.

Event Management

Enables organizations to identify service health-related issues on a single management console. Event Management provides alert aggregation and root cause analysis (RCA) for discovered services, application services, and automated alert groups.

Service Portfolio Management (Service Portfolio Management)

Enables organizations to document and manage services using a standardized, structured format.

Products that benefit from CSM

IT Service Management (ITSM)

Enables organizations to link incidents, problems, changes, and requests to cases, and have the context of the customer (consumer or account) reporting the issue.

IT Operations Management (ITOM)

Enables organizations to identify the Install Base Items and the customers affected by service issues. Helps organizations to provide proactive customer service.

Service Portfolio Management (Service Portfolio Management)

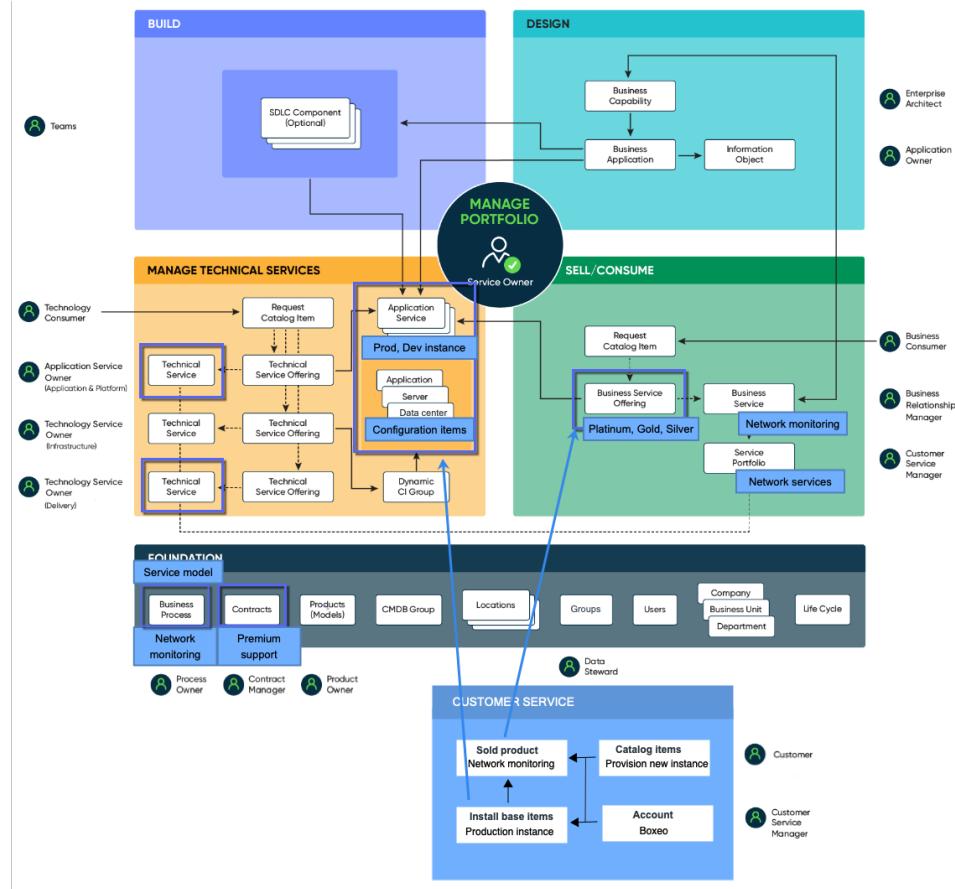
Enables customers that have subscribed to the Service Offering to see who owns the service.

In this example, the CSDM helps the fictitious ACME technology company use CSM to manage network monitoring services for customers.

Example scenario: Key personas and how they benefit

ACME offers a network monitoring service to its enterprise customers who can purchase either the Platinum, Gold, or Silver offering. A customer, Boxeo, has purchased the network monitoring Platinum offering (Sold Product) and is using it in development and production environments (Install Base Item).

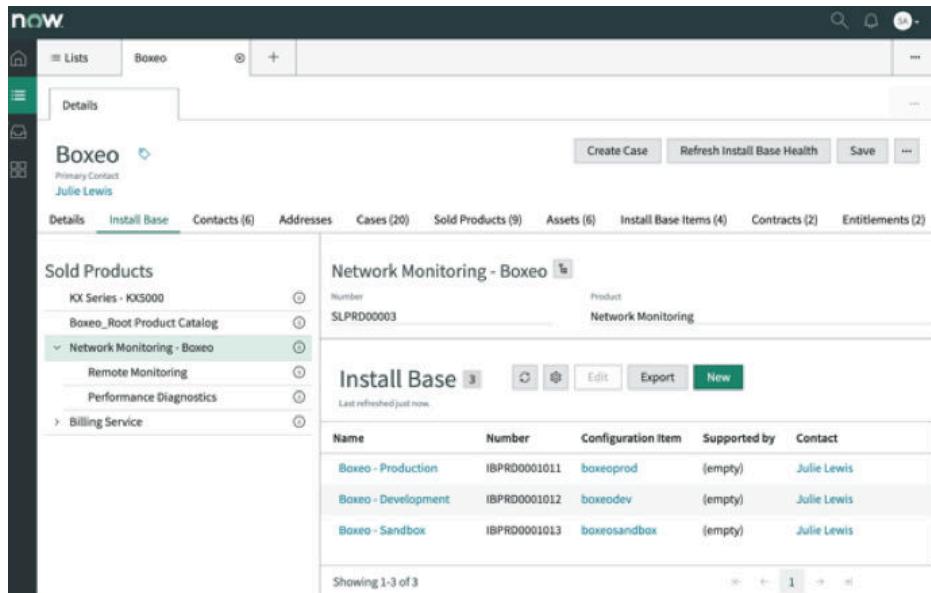
CSM data model example



The key personas (customer service agents, network operations center (NOC) engineers, service owners, and customers) can complete the following tasks:

Customer service agents can:

- View the products and services that the customer has purchased (Sold Product) and installed (Install Base).



Sold Products

Number	Product
SLPRD00003	Network Monitoring

Install Base

Name	Number	Configuration Item	Supported by	Contact
Boxeo - Production	IBPRD0001011	boxeoprod	{empty}	Julie Lewis
Boxeo - Development	IBPRD0001012	boxeodev	{empty}	Julie Lewis
Boxeo - Sandbox	IBPRD0001013	boxeosandbox	{empty}	Julie Lewis

- View the service offering associated with the sold product. The sold product references the service offering.

The screenshot shows the ServiceNow interface for a 'Platinum Network Services' item. At the top, there are tabs for 'Details', 'Network Monit...', and 'Platinum Netw...'. Below the tabs, the item's details are listed: Class, Owned by, Status, Updated, Service Offering (Alex Linde), Installed (2020-05-19 00:00:06). A 'Save' button is visible. The main content area is titled 'Service Commitments (3)'. It lists three items with their descriptions and values: '99.99% Availability 24X7' (100), '24 X 7 Support' (200), and 'Nightly Backup' (300). There are buttons for 'New', 'Create Case', 'Refresh Install Base Health', and 'Save'. At the bottom, it shows 'Showing 1-3 of 3' and a '20 rows per page' dropdown.

- View the health status of Install Base Items

The screenshot shows the ServiceNow interface for the 'Boxeo' workspace. At the top, there are tabs for 'Details', 'Install Base', 'Contacts (6)', 'Addresses', 'Cases (12)', 'Assets (5)', 'Install Base Items (5)', 'Contracts (3)', 'Entitlements (2)', and 'Accounts (3)'. The 'Install Base Items (5)' tab is selected. The main content area is titled 'Install Base Items (5)'. It lists five items with their names, numbers, configuration items, health statuses, and other details. The items are: 'KX Series - KX5000' (IBITM0000901, KX Series - KX5000, Not Available, 2020-07-29 12:20:32, (empty), Julie Lewis), 'Boxeo - Rewards Service' (IBITM0001001, Rewards Service, Normal, 2020-07-29 12:20:32, Rewards Service, Julie Lewis), 'Boxeo - Production' (IBITM001001, boxeoprod, Critical, 2020-07-29 12:20:32, Network Monitoring, Julie Lewis), 'Boxeo - Development' (IBITM001002, boxeodev, Major, 2020-07-29 12:20:32, Network Monitoring, Julie Lewis), and 'Boxeo - Sandbox' (IBITM001003, boxeosandbox, Normal, 2020-07-29 12:20:32, Network Monitoring, Julie Lewis). There are buttons for 'New', 'Create Case', 'Refresh Install Base Health', and 'Save'.

NOC engineers can: View customers affected by a service issue and inform customer service. Specifically, they can perform the following operations:

- View the affected Install Base Items and add it to or remove it from an Account. The Install Base item references the application service (CI). The CI it depends on should be one of the CIs affected. The CI should also be referenced in the Alert to show that Install Base Item (and therefore the Account or Consumer) as affected.
- Create a proactive case from an alert and inform the customer service team of the service issue or outage.

The screenshot shows the ServiceNow interface with the title 'Alert2020119'. Below the title, there's a section titled 'Affected Install Base Items [6]'. A table lists six items, each with a 'Number' (e.g., IBITM0001001), 'Account' (e.g., Boxeo), 'Name' (e.g., Boxeo PROD), and 'Configuration Item' (e.g., Rewards Processing).

Number	Account	Name	Configuration Item
IBITM0001001	Boxeo	Boxeo PROD	Rewards Processing
IBITM0001001	Boxeo	12000XHD Digital Press IB	12000XHD Digital Press
IBITM0001002	Avid Corporation	Avid PROD	Rewards Processing
IBITM0001002	Diagonal Inc.	Diagonal Inc PROD	Rewards Processing
IBITM0001003	Advances Super Computing	Advanced Super Computing PROD	Rewards Processing
IBITM0001003	Boxeo	Monitoring Service for Boxeo	boxeoprod

Service owners can: View the customers that are subscribed to a service offering.

The screenshot shows the ServiceNow interface with the title 'Platinum Network Services'. Below the title, there's a section titled 'Subscribed by Customers [2]'. A table lists two customers, 'Network Monitoring - Boxeo' and 'Network Monitoring', each associated with 'Network Monitoring' products and 'Boxeo' accounts.

Name	Product	Account
Network Monitoring - Boxeo	Network Monitoring	Boxeo
Network Monitoring	Network Monitoring	Avid Corporation

Customers can:

- View the install base items and details.
- View outages and service issues.

The screenshot shows the ServiceNow interface for ACME CORPORATION. The top navigation bar includes links for Knowledge, Requests, My Lists, Case, Support, Notification (with 11 notifications), Tours, and a user profile for Julie Lewis. The main content area displays an 'Outage - Service Unavailable - Network Monitoring' record for Boxeo - Production. Below it is a 'Cases from last 30 days' grid. To the right, there are 'Related Actions' like 'Create Case' and sections for 'Products on this Instance' (Payment Processing, Network Monitoring - Boxeo) and 'Billing Service' (Billing Service Model). A 'Service Status from last 30 days' chart is also present.

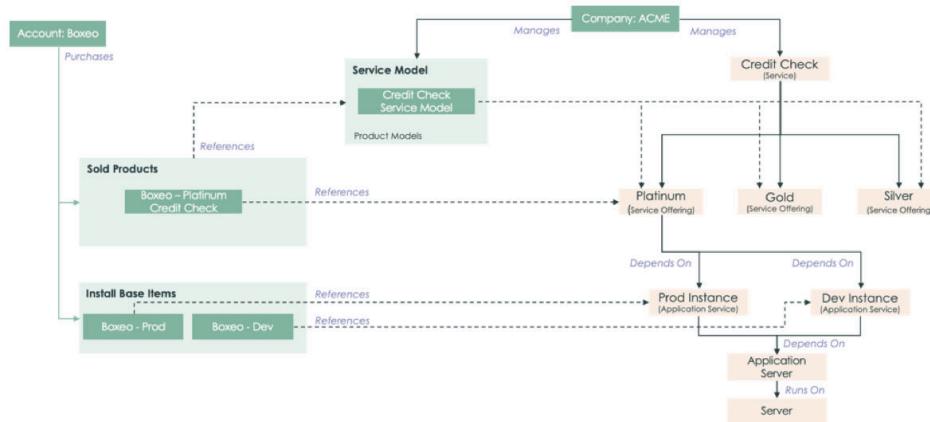
- View products they have purchased.
- Request services related to the products they have purchased.

The screenshot shows the ServiceNow interface for ACME CORPORATION. The top navigation bar is identical to the previous screenshot. The main content area displays a 'Product Details' view for Network Monitoring. It shows the product name, account (Boxeo), and model categories (Application Service). Below is a 'Cases from last 30 days' grid. To the right, there are 'Related Actions' like 'Create Case', 'Service Catalogs' (Provision New Instance, Remove Demo Data), and 'Install Base' (listing instances for Boxeo - Sandbox, Boxeo - Production, and Boxeo - Development).

Example CSM and CSDM use cases

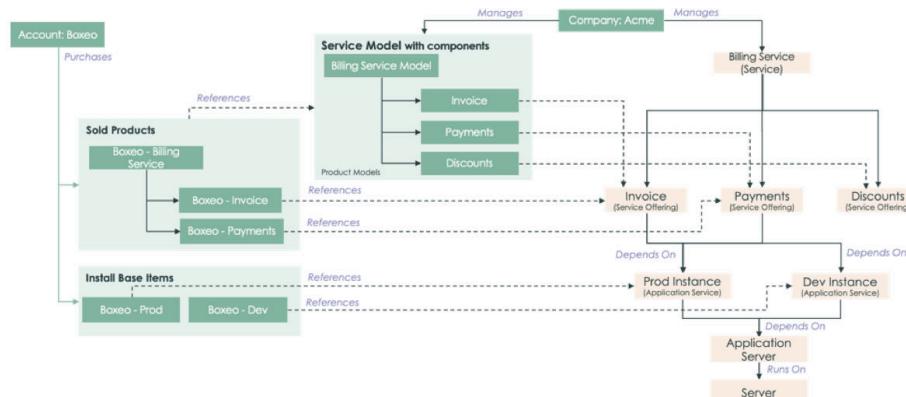
Service with multiple packages

ACME offers a Credit Check Service to its enterprise customers. They can purchase the Platinum, Gold, or Silver offering.



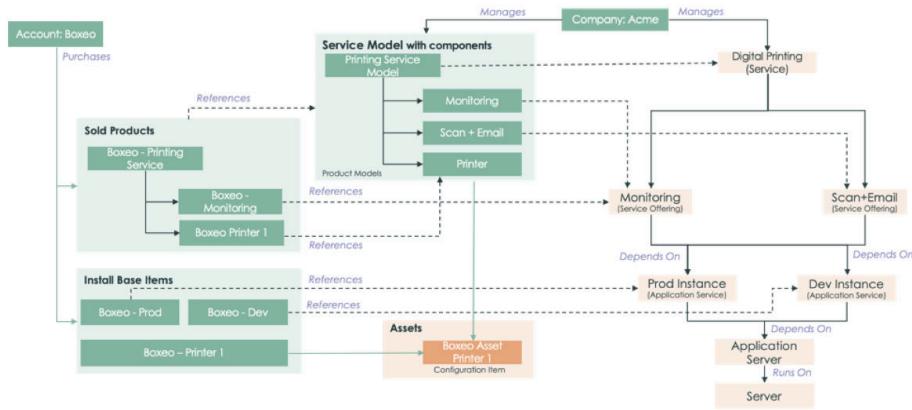
Service with optional components

ACME offers a Billing Service to its enterprise customers. They can purchase the bundle or one or more of the component offerings (for example, invoices, payments, or discounts).



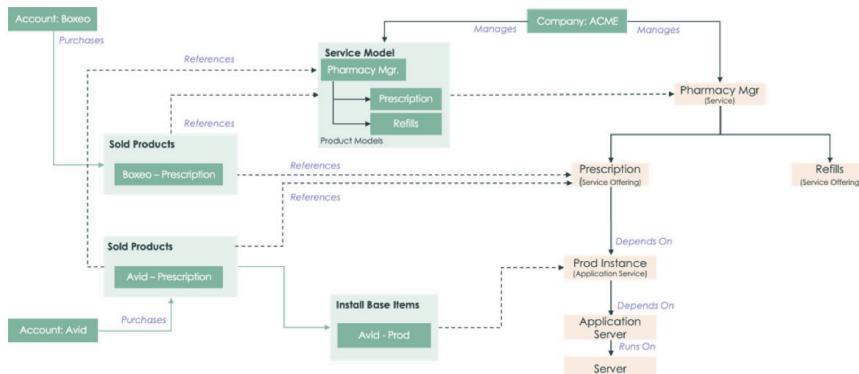
Service with a physical product

ACME offers a Digital Printing Solution. Customers can purchase either the bundle or the printer along with one more service offerings (for example, Scan and Email, Monitoring).



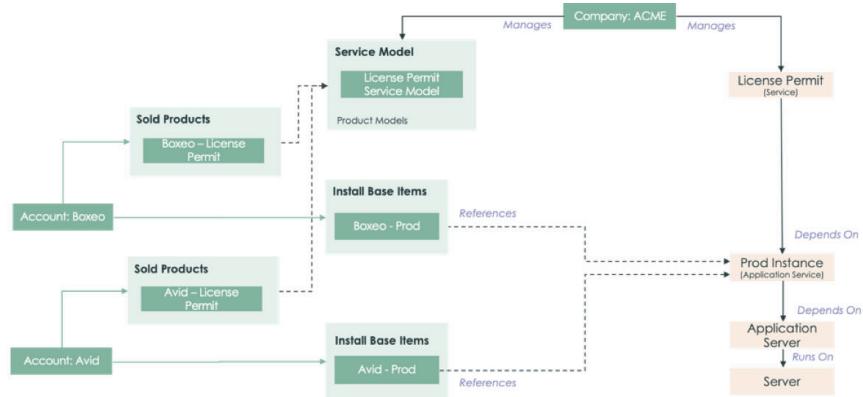
Service sold to multiple customers

ACME sells a Pharmacy Manager Service to two customers: Boxeo and Avid Inc.



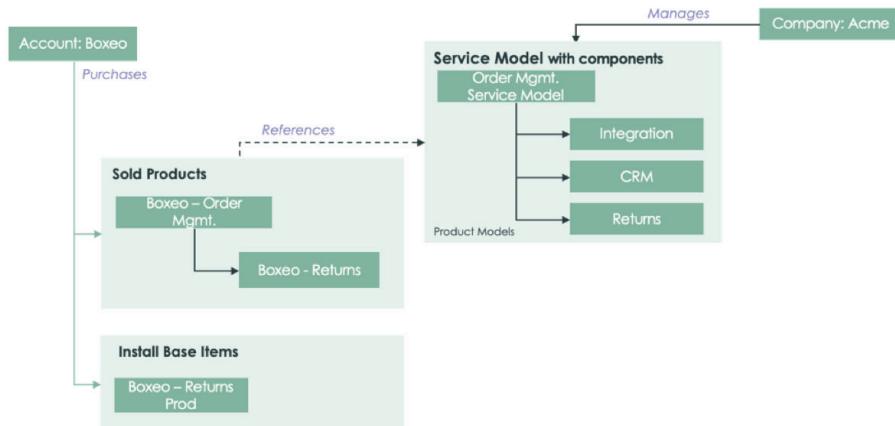
Service used by multiple customers

ACME deploys both Boxeo and Avid on the same production instance (multi-tenant model).



Service used on-premises

ACME offers an Order Management service. This purchase is tracked in the Now Platform but is used on-premises.



Additional information

For more information about the relevant CSM features and tasks, see the following topics:

[Configure form views for Service Portfolio Management integration](#)

[View product information from the Customer Service Portal](#)

[Create a proactive case from an alert](#)

Create a case for install base from the Customer Service homepage

Proactive Customer Service Operations

Service health status for install base

Configure install base

Outage tracking for install base

View install base information from the Customer Service Portal

View install base information in Agent Workspace

View product information from the Customer Service Portal

View sold product information in Agent Workspace

Consider these points while implementing the CSDM framework.

Frequently asked questions (FAQs)

- What is a Service-aware Install Base?

A Service-aware Install Base enables companies to track the digital products and services in use. A Service-aware Install Base also tracks the relationships of the products and services to dependent services and CLs that affect their health.

- What are Proactive Customer Service Operations?

Proactive Customer Service Operations bring CSM and Event Management together to enable companies to proactively trigger case workflows and notify the affected customers.

- Do I need to purchase CSM Professional package to use the Service-aware Install Base?

No. The Service-aware Install Base is included in the CSM base system.

- Can multiple sold products reference the same service offering?

Yes. Multiple sold products (that is, Service Models) purchased by different companies can reference the same service offering. For

example, multiple customers can purchase the same SaaS offering with same service commitments.

- Can multiple Install Base Items reference the same application service?

Yes. Multiple Install Base Items (either for the same account or for different accounts) can reference the same application service. For example, a multi-tenant SaaS offering where multiple customers (each with their own Install Base Item) are used on the same production instance (application service).

- When do I create an incident rather than a proactive case from an alert?

Typically, some companies create an incident so that their NOC engineers can resolve the issue. After they determine that the issue impacts multiple customers, they also proactively create a major case and related child cases (one for each impacted customer) to notify the affected customers. Thus, the alert, incident, and case are all linked.

Updates made by the resolving teams to the incident status or additional comments are reflected in the case. The customer service teams use these updates to keep customers informed.

To meet the customer notification time requirements of the SLAs, companies may also automate creating incidents and cases from the alert. In addition, companies can also create a proactive case from the alert while the issue is being resolved.

- What is the difference between entitlements managed at the CSDM Service Offering and the CSM Contracts and Entitlements?

Service Commitments in CSDM define the expected level of a service. A service offering consists of a set of service commitments which uniquely define the service offerings. For example, a service offering may include a service commitment to perform a data backup each night.

Service contracts in CSM store information about the type of support that is provided to a customer. A contract can include an account or consumer, a contact, and the specific assets that are covered. A contract can also include multiple service entitlements and SLAs. An entitlement defines the type of support that a customer receives, as well as the supported communication channels. For example, a

customer may sign a service contract to receive support from 6:00 a.m. to 9:00 p.m. on weekdays.

- How can I request additional services based on the product that I've purchased?

The relationship between the product model and the catalog items enables you to use the customer portal to request additional services for the products you've purchased. Multiple catalog items can be associated with a product model.

- Do I need elements from all the CSDM domains to set up CSM?

No. The approach mentioned in this use case is based on the recommended guidelines and assumes you are in the Run or Fly stage of the CSDM implementation. See [Implementing the CSDM framework in stages](#) for more information.

When you are implementing CSM, start with the tables in the CSDM Foundation domain and the CSM Customer Service domain. Using these domains enables you to leverage the capabilities included in CSM.

To enable the proactive customer service operations, use tables from the CSDM Manage Technical Services domain for monitoring the application services tied to the customers' install base. Using this domain enables you to leverage the CSM and ITOM integration.

Service-centric organizations can leverage the tables from the CSDM Sell/Consume domain to connect the product model to the service offerings and then to sold product. These connections enable you to track the service portfolio and see a complete view of how customers are consuming services.

DevOps Config product view

The DevOps Config application validates and manages the configuration data of your enterprise applications across every stage of the DevOps pipeline. This ensures that risky changes to configuration data do not get deployed to your production environment. The goal of this product view is to help you to understand how DevOps Config key entities work with the core CSDM framework.

With DevOps Config, you can:

- Automatically validate configuration data before deployment.
- Integrate your most common DevOps tools and processes.
- Evaluate and monitor DevOps Config process and results.
- Manage and secure your configuration data across multiple sources.
- Use DevOps Config with DevOps Change Velocity for visibility across the entire life cycle of your applications.

For more information

For more information, see [Configuring DevOps Config](#)

- [DevOps Config and CSDM tables](#)

DevOps Config manages and uses CSDM tables. Several ServiceNow products benefit from and add value to DevOps Config.

- [DevOps Config use case](#)

DevOps Config centralizes configuration data so that it can be secured and validated before deploying to production. DevOps Config supports continuous deployment processes by validating configuration data changes for use downstream by deployment tools. Governance teams can use policies to help developers deliver compliant products with minimal impact on the pipeline.

- [DevOps Config considerations](#)

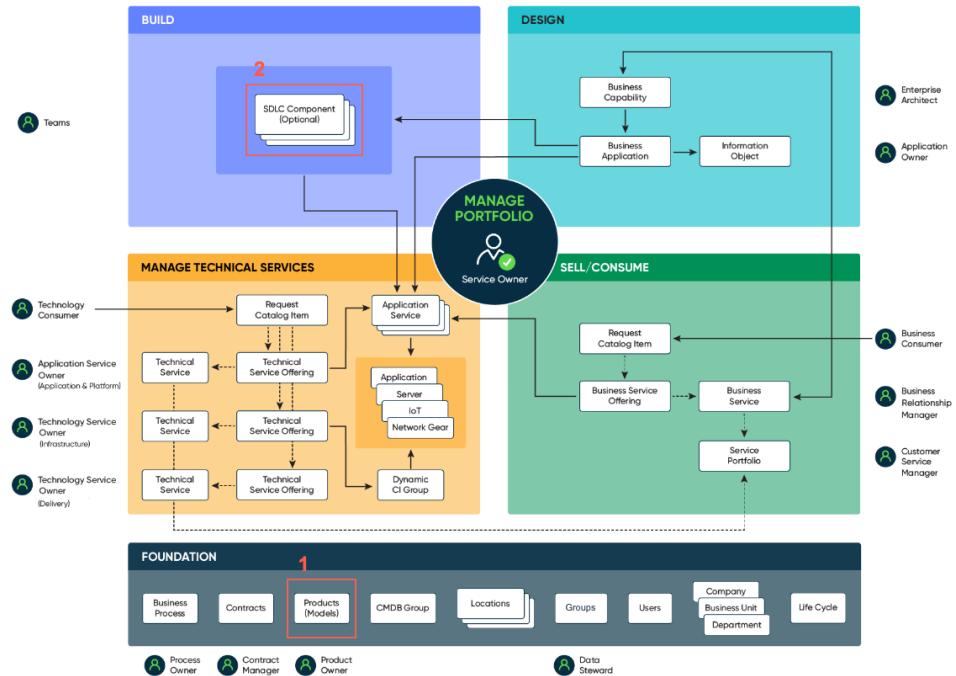
Consider these points while implementing the CSDM framework.

DevOps Config manages and uses CSDM tables. Several ServiceNow products benefit from and add value to DevOps Config.

CSDM tables managed by DevOps Config

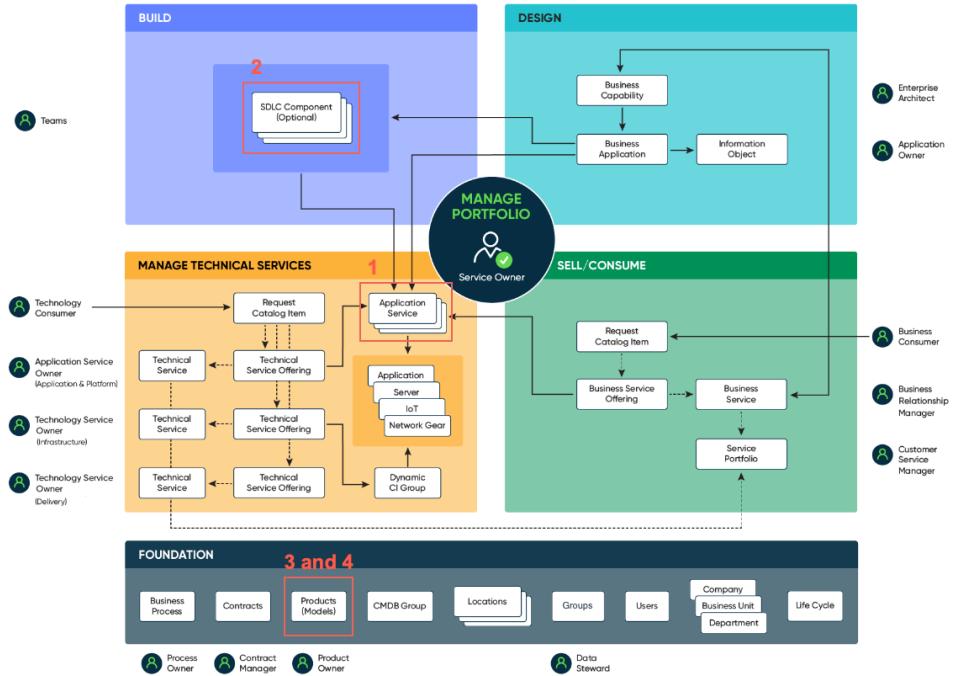
The following tables are managed by DevOps Config:

- Application Model table [cmdb_application_product_model]
- SDLC Component table [cmdb_ci_sdlc_component]



CSDM tables used by DevOps Config

- Application Service table [cmdb_ci_service_auto]
- SDLC Component table [cmdb_ci_sdlc_component]
- Application Model table [cmdb_application_product_model]: Extends Product Model table
- Software Model table [cmdb_software_product_model]: Extends Product Model table



Products that add value to DevOps Config

- Change Management captures configuration data changes along with application code changes, test summaries, and other DevOps related points attached to a change request record.
- DevOps Change Velocity integrates configuration data into pipeline executions.

Products that benefit from DevOps Config

- ITSM: Applications show configuration data as a part of changes in their application's life cycle, which can be seen in pipeline executions and change request records.
- GRC: You can map control objectives to configuration data policies to provide an audit trail of configuration data changes in real-time for risk and compliance use cases.

DevOps Config centralizes configuration data so that it can be secured and validated before deploying to production. DevOps Config supports

continuous deployment processes by validating configuration data changes for use downstream by deployment tools. Governance teams can use policies to help developers deliver compliant products with minimal impact on the pipeline.

Key features of the DevOps Config use case

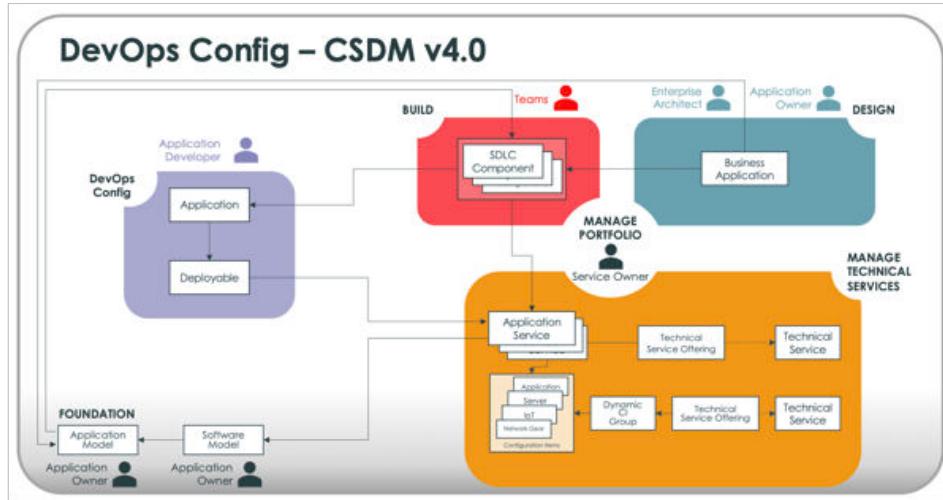
- Manage configuration data in a secure, centralized data model.
- Leverage out-of-the-box policy content for faster adoption.
- Harness compliant configuration data in the pipeline.

DevOps Config model with CSDM

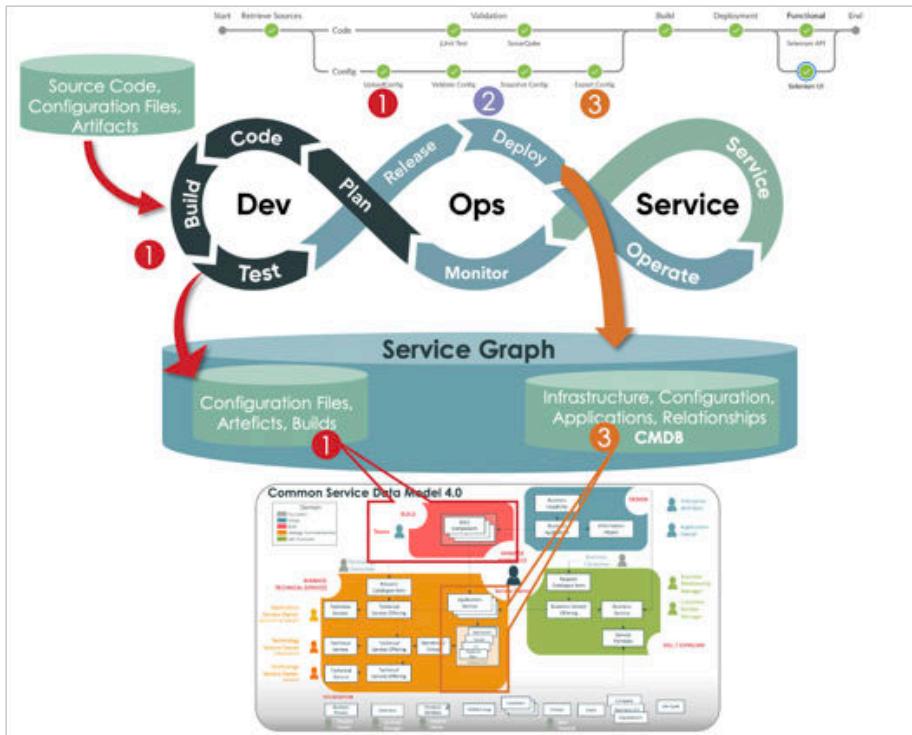
DevOps Config ties to the Common Service Data Model through two different points, emphasizing both the Build and Operate aspects of the model:

- In the Build phase, DevOps Config connects via the CDM Application table to the SDLC Component table, which then connects to the application model. This allows DevOps Config teams to manage the overall configuration data for their applications and infrastructure.
- DevOps Config also interacts with CSDM in the Application Service table. DevOps Config ties a CDM application's deployable(s), which is the set of configuration data that is deployed to a particular environment such as production, development, or QA to an application service. The application service is representative of an application's operating environment, including environment type and location attributes which maps perfectly with the CDM deployable object.

Here is a focused view of the CSDM diagram, with the parts that are relevant to DevOps Config.



Additionally, the following diagram shows how configuration data flows operationally through DevOps Config and where it touches the various elements of CSDM.



1. Upload

- When an application is built, it is usually handled by an automated build service like Jenkins. In addition to the application code, configuration data is also pulled into the build service. Users can upload the configuration data into DevOps Config and commit it to their application's data model. The data is linked to an application model through the SDLC component.
- Additionally, during the commit stage, if configuration data changes are found to impact the application's deployables, a snapshot is generated to capture this change for that environment.

2. Validate and publish: Configuration data changes that generate snapshots for particular deployables can be validated against a user's policies. After a snapshot is validated, it can be published for consumption.

3. Export:

- a. After the configuration data is published, it can be exported. The exported configuration data is used downstream in a CI/CD pipeline, where it is used to provision an application or infrastructure.
- b. If DevOps Change Velocity is used along with DevOps Config, users can implement change acceleration in this step of the pipeline to link a snapshot to a change request. Additionally, the application service linked to the deployable that the snapshot belongs to can also be specified. If the application service is appropriately linked to a Dynamic CI Group, it can create a direct link between the CIs in the application service that would be impacted by the changes specified in the snapshot.

Results of the DevOps Config use case

CSDM provides DevOps Config connectivity between application models for building and ensuring that the instances of that model that are running in a customer's environment (as expressed in the Application Services) are using a validated configuration data in the build and deploy processes for the application and/or infrastructure CIs tied to that environment.

Consider these points while implementing the CSDM framework.

- If you are creating a new DevOps Config application from an existing application model or service, consider the differences between an Application Model and an Application Service:
 - An Application Model is a type of model that can be managed and tracked within a Scrum development process. It is the reference object to which your application's configuration data will be tied.
 - An Application Service represents an instance of a business application or system in different types of development environments, like development, test, or production. It is the target environment where your configuration data will be deployed and is consequently linked to a deployable in your application. Application Services typically link back to an application model, where the intended state of configuration data for an application is defined.

Note:

- If you already have a target environment (development, test, or production) that you want to configure and the Application Service for this environment has been defined, this option is the quickest way to get your configuration data defined and mapped to that environment. It's the preferred option.
- This is also the preferred option if an Application Service has not yet been defined, but an application model has already been created (from other uses in the platform).
- Usage of a separate application record in DevOps Config:

DevOps Config has a separate version of an application compared to its counterpart in DevOps Change Velocity. The reason for this is that the application is the central object of reference for a customer's configuration data model, and therefore requires strict boundaries on it to respect any changes to that model over time.

Conversely, the application in DevOps Change Velocity can be considered more of a container for the various tool objects that can be discovered with it (pipelines, plans, repositories, and code commits).

Incident Management product view

Incident Management supports the incident management process with the ability to identify and log incidents, classify and prioritize incidents, assign incidents to appropriate users or groups, escalate, resolve, and report incidents. The goal of this product view is to help you to understand how Incident Management key entities work with the core CSDM framework.

Features of Incident Management that get the most benefit from the CSDM include:

- Agent Workspace gives agents the information they need to quickly prioritize and resolve incidents.
- The major incident workbench includes a single-pane view you can use to identify, track, and resolve high-impact incidents.

- The native mobile app allows agents to quickly view and respond to tasks on-the-go, and can approve the requests with a single swipe.
- Incident deflection encourage self-help by suggesting related knowledge base articles.
- Improves collaboration on incident tasks by using drag-and-drop functionality on visual task boards.
- Performance analytics provide detailed insights into performance trends.

Incident Management includes a form you can use to report incidents.

Incident form

The incident form references the following attributes and related lists.

1. Service — References the [cmdb_ci_service_business] table.

Note: Earlier ServiceNow releases labeled this attribute Business Service.

2. Service Offering (attribute) — References the [service_offering] table. Displays the service offerings affected by the incident in the Service Offerings related list.

3. Configuration Items — References the [cmdb_ci] table

4. Affected/Causal CIs — Related list [task_ci] table. (The Incident form allows Application Services to be chosen as CIs)
5. Impacted Services — Related list [task_cmdb_ci_service] table
6. Service Offering — Related list [task_service_offering] table

For more information

For more details about Incident Management, see [Incident Management](#).

- [Incident Management and CSDM tables](#)

Incident Management manages and uses CSDM tables. Several ServiceNow products benefit from and add value to Incident Management.

- [Incident Management use case](#)

Incident Management restores normal service operation, while also minimizing impact to your business and maintaining the quality of your data.

- [Set up the Incident Management form](#)

Configure the Incident form to see the impact of an incident and then restore affected services. The CSDM framework enables you to view rich context for incidents: the CIs involved in the incident and the service offerings, business applications, and business services that the incident affects.

- [Incident Management considerations](#)

Consider these points while implementing the CSDM framework.

Incident Management manages and uses CSDM tables. Several ServiceNow products benefit from and add value to Incident Management.

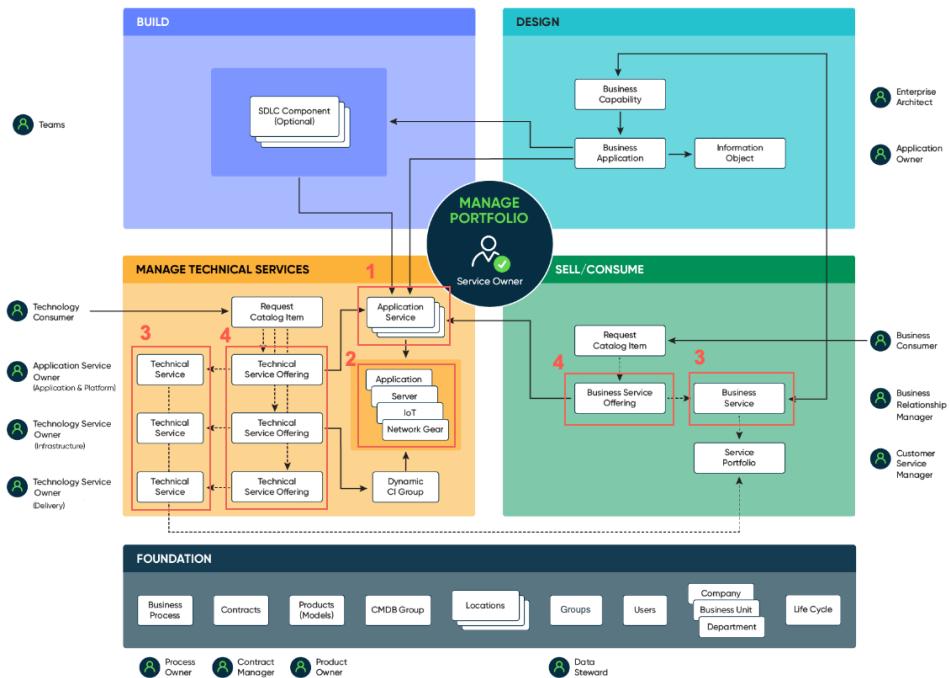
CSDM tables used by Incident Management

The Incident form references the following CSDM tables:

1. Application Service [cmdb_ci_discovered_service] table or any infrastructure CI.
2. Configuration Item tables [cmdb_ci*]
3. Business Service [cmdb_ci_service_business] table and Technical service [cmdb_ci_service_technical] table: Use the service classification attribute to identify business services, technical services, and application services as types of services.
4. Service Offering [service_offering] table: Uses the service classification attribute to identify business services, technical services, and application services as types of service offerings.

See the Incident Management documentation for information about service commitments on a service offering (for example, resolution time for incident).

Tables used by Incident Management



Products that add value to Incident Management

When you use Incident Management with other ServiceNow products, you increase the value you get from Incident Management. These other ServiceNow products include:

Discovery

Discovery provides details about the hardware and software CIs you are using.

Service Portfolio Management (Service Portfolio Management)

Provides life-cycle information for a service.

Asset Management

Provides the related product model. Software Asset Management (SAM Foundation) and Hardware Asset Management (HAM) provide lifecycle data for Incident Management.

Security Management

Provide initial information that helps in containing, eradicating, and recovery of security related incidents.

Risk Management

Provides IT risk and financial risk information.

Products that benefit from Incident Management

IT Service Management (ITSM)

Services have the context of the business and applications, along with the information and technologies layered beneath them.

Information Technology Operations Management (ITOM)

Understands the business context for the application services along with the hardware and software being managed.

Governance, Risk, and Compliance (GRC)

Auditors can leverage the business applications and related Information Objects. This helps auditors understand the design-time data sensitivity for scoping audits, measuring risks, and managing audit activities.

Asset Management

Manages the software and hardware life cycles for business applications and business services.

Customer Service Management (CSM)

Incident Management restores normal service operation, while also minimizing impact to your business and maintaining the quality of your data.

Incident Management Use Case

Use Incident Management to create an incident that captures information about the asset-related CIs. An incident keeps a record of the updated, repaired, swapped, or retired CIs.

By keeping track of the assets, you can tell where the assets are located, how they are used, and when changes were made to them. This information helps you systematically monitor and manage the assets in your company.

A CI generates an incident. Use a dependency view to identify other CIs that are affected by the incident. Associate the affected CIs with an incident record to find out how the incident affects other dependent CIs.

Following the CSDM framework provides value to Incident Management in the following ways:

- Incident Management understands the impact of the Incident on services and service offerings.
- Incident Management dynamically routes incidents.
- Incident Management identifies all affected services.

Results of the use case

The CSDM framework provides context for incidents—both the CIs involved in the incident and the services that it affects.

Use the Incident form to see the impact of an incident and restore affected services. Complete the following steps:

1. Populate the **Configuration Item** attribute [configuration_item] with the CI or the affected service. You can then use the CI to identify details for incident routing. For example, you can use the CI data like Support Group and provide information about the service impact by using dependency relationships.
2. Populate the **Impacted Services** related list [task_cmdb_ci_service] with the services and service offerings that are related to the populated CI.
3. (Optional) Use the **Service** and **Service Offering** attributes to help narrow the list of available CIs.

Note: Narrowing the list of available CIs is not a feature of the base system. To narrow the list, you need to configure the Incident form.

4. (Optional) Add the **Affected CI** related list [task_ci] to identify the CIs that might have caused the incident.
5. (Optional) Add the **Impacted Services** related list [task_cmdb_ci_service] to see the services and CIs that are impacted by the incident.

Incident form for use case

The screenshot shows the ServiceNow Incident form for use case. The top half of the form contains fields for Number (INC0010006), Caller (Praveen Vankar), Category (Software), Subcategory (Email), Business service (Email and Calendaring), Service offering (Email - Cloud offering), and Configuration item (Configuration Item). The bottom half includes a Short description field and a large Description text area. Below the form is a Related Records panel with tabs for Notes, Resolution information, Problem, Change Request, and Caused by Change. At the bottom is a Related Links section with Task SLAs, Affected CIs, and Impacted Services/CIs. The Affected CIs and Impacted Services/CIs buttons are highlighted with red boxes and numbered 4 and 5 respectively.

Configure the Incident form to see the impact of an incident and then restore affected services. The CSDM framework enables you to view rich context for incidents: the CIs involved in the incident and the service offerings, business applications, and business services that the incident affects.

Before you begin

Role required: admin

About this task

Incident Management leverages the CSDM data structure to display information that assists in solving the incident.

[Go to the 1:00:00 mark on this video to learn about setting up the Incident form to include the related lists that leverage the CSDM framework.](#)

Procedure

1. Navigate to **All > Incident > Administration > Incident Properties**.
2. In the Incident Related List Properties section, enable the following properties and then select **Save**.
 - **Populate Impacted Services based on Affected CIs.** This property enables display and update of the **Impacted Services/CIs** related list when you perform the **Refresh Impacted Services** action.
 - **Populate the Business Application related list for incidents**
 - **Populate the Service Offering related list for incidents**

The properties are fully described in [Incident Management properties](#).

3. Ensure that the **Principle Class** option is selected for the CI or CI class as described in [Create a CI class](#) and [Update the list of classes in the Principal Class filter](#).
4. Navigate to **All > Incident > Open** and fill in the following fields:

Incident form

Service	<p>The business service or technical service associated with the incident. The value that you specify narrows the list of available CIs that appear for selection in the Configuration Item field.</p> <p>Note: Narrowing the list of available CIs isn't a feature of the base system. To narrow the list, you must specify values for Service and Service Offering.</p>
---------	---

Service Offering	<p>The business service offering (product) or technical service offering associated with the incident. The value that you specify narrows the list of available CIs that appear for selection in the Configuration Item field.</p> <p>Note: Narrowing the list of available CIs isn't a feature of the base system. To narrow the list, you must specify values for Service and Service Offering.</p>
Configuration Item	<p>The CI or the affected service. (Hardware, application, and cloud [physical CIs] or application service [logical CIs])</p> <p>You can then use the CI to identify details for incident routing. For example, you can use the CI data like Support Group and provide information about the service impact by using dependency relationships.</p>

The screenshot shows the ServiceNow Incident management interface. At the top, there are fields for Number (INC0010006), Caller (Proven Veldkamp), Category (Software), Subcategory (Email), Business service (Email and Calendaring), Service offering (Email - Cloud offering), and Configuration item (No work). Below these, there are sections for Contact type (Phone), State (New), Impact (3-Low), Urgency (3-Low), and Assignment group (Priority). The 'Notes' tab is selected, followed by 'Related Records' and 'Resolution Information'. Under 'Related Links', there are two tabs: 'Affected CIs' (highlighted with a red box) and 'Impacted Services/CIs' (also highlighted with a red box). A legend at the bottom defines icons for Task, Task List, Configuration Item, Managed by, Denied by, Approval group, Location, Operational status, and Manually added.

5. Review the information.

- The CIs that may have caused the incident appear in the **Affected CIs** related list [task_ci].
- The services and service offerings that are related to the populated CI appear in the **Impacted Services/CIs** related list [task_cmdb_ci_service].
- The services and CIs affected by the incident appear in the **Impacted Services/CIs** related list [task_cmdb_ci_service].

Consider these points while implementing the CSDM framework.

Incident Management considerations

Business applications are not referenced in Incident Management

Business applications are portfolio objects used for designing and planning your enterprise architecture. Business applications don't contain attribute-level details such as version, environment, and localization (for any deployments using one or more applications).

Application service

An application service is an operational CI and a unique instance of an application.

An application service is the logical representation of the underlying hardware and software CIs that work together to implement a business application or system. The application service represents an instance of the business application or system.

Using application service on an Incident form

You can use the application service CI on the Incident form in scenarios like the following:

- The incident is for an application-related issue. On the Incident form, you can enter Application Service in the **Configuration Item** field to represent the application. For example, you can report the application service called MyApp 3.0 Production as unavailable.
- The incident is for an infrastructure CI that is affecting one or more services. On the Incident form, the **Impacted Services/CIs** related list identifies the application service affecting services. For example, the Server Acme42 CI might be identified as affecting the MyApp 3.0 Production and other related services.

Operational Technology product view

Operational Technology covers products that tackle aspects of managing OT assets and production processes at various stages of the life cycle. The goal of this product view is to help you to understand how Operational Technology key entities work with the core CSDM framework.

Operational Technology Manager

Creates the foundational data and relationships that enable your enterprise to use the Operational Technology solution. For more information, see [Operational Technology Manager](#).

Industrial Process Manager

Enables you to create the ISA-95 Equipment Model data foundation that is required for the Operational Technology solution. For more information, see [Industrial Process Manager](#).

Operational Technology Vulnerability Response

Enables effective prioritization and remediation of OT asset vulnerabilities at the site level. For more information, see [Operational Technology Vulnerability Response](#).

Operational Technology Incident Management

Enables engineers to quickly resolve OT asset and production process issues. For more information, see [Operational Technology Incident Management](#).

Operational Technology Change Management

Enables your organization to implement changes to OT assets and production processes. For more information, see [Operational Technology Change Management](#).

- [Operational Technology and CSDM tables](#)

Operational Technology manages and uses CSDM tables. Several ServiceNow products benefit from and add value to Operational Technology.

- [Operational Technology Manager use case](#)

The Operational Technology Manager use cases are described in this section.

- [Operational Technology and CSDM elements](#)

Terms related to managing business applications with elements of CSDM.

- [Operational Technology FAQ](#)

You might have questions while implementing the CSDM framework.

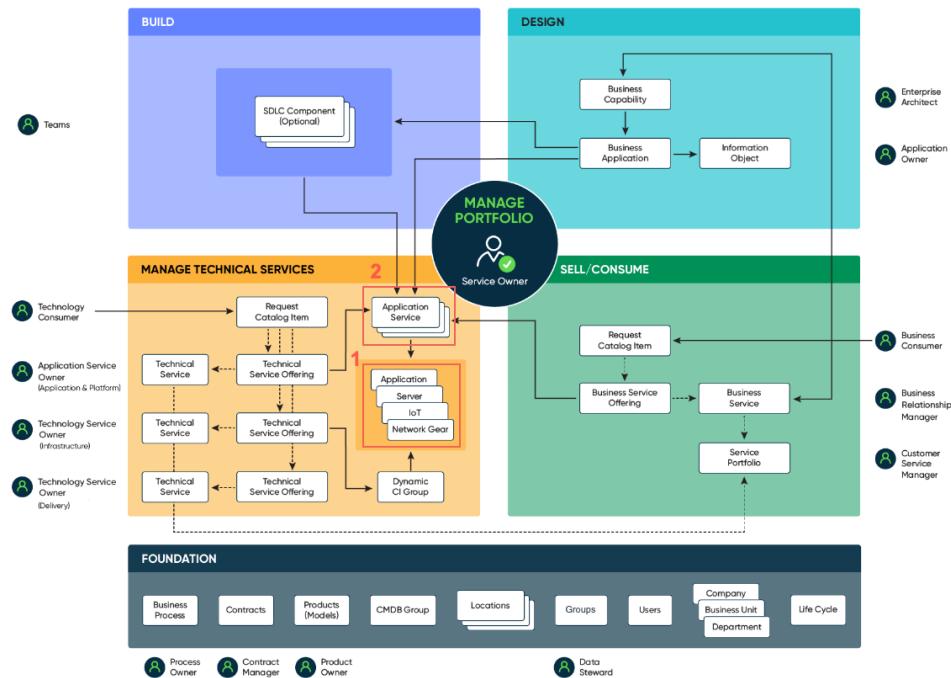
Operational Technology manages and uses CSDM tables. Several ServiceNow products benefit from and add value to Operational Technology.

CSDM tables managed by Operational Technology

There are two primary categories of tables managed by Operational Technology (OT):

- Operational Technology assets: Configuration items found on an OT (ICS or PCN) network.
- ISA equipment model entity: Industrial process automated by OT assets.

The numbers in this figure correspond to the CSDM tables managed by Incident Management.



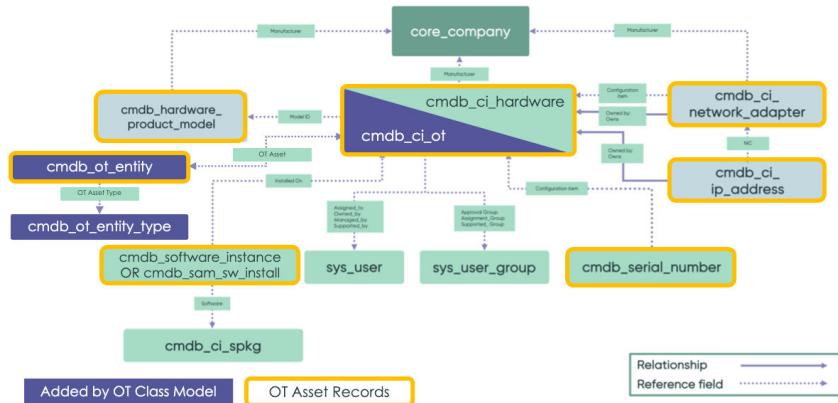
CSDM tables used by Operational Technology

1. OT assets:

- a. Configuration Item classes were created for Operational Technology hardware classes (cmdb_ci_ot) by extending hardware. See [Operational Technology \(OT\) extension classes](#) for details.

- b. Any CI Class (any relevant existing hardware class as well as new OT classes can be designated as OT assets by adding OT asset details using the OT Asset Details (cmdb_ot_entity) reference to the cmdb_ot_entity table. OT Asset Details include OT-specific characteristics like Purdue Level and OT asset type.
- c. OT asset types describe the function of any CI that automates an industrial or production process. The cmdb_ot_entity_type table describes these functions or roles.

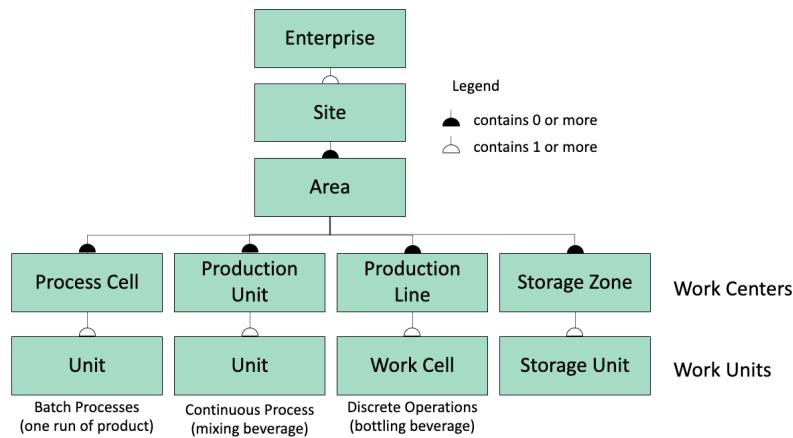
As shown here, a single OT asset is represented by at least two records: one CI and one OT entity record. The asset can contain six or more records in up to six tables (for example, if the CI has more than one IP and MAC address).



2. Equipment model entities:

- a. The equipment model entity class extends the Calculated Application Service and is used to:
 - a. Represent the site of any OT asset or equipment model entity. A record in the Equipment Model Entity table (cmdb_ci_ot_isa_entity) without a parent is considered a site.
 - b. Represent the ISA equipment model entity for a part of the production process.
- b. You can use equipment model templates (isa_entity_template) to further describe the relationships between equipment model entities found in an industrial environment.

- a. Levels (`isa_entity_level`) describe the hierarchical level of the equipment model entity. For the default ISA-95 template, the levels shown here (area, work center, and work units) are included in the base system.
- b. Level types (`isa_entity_type`) describe the type of process represented by the equipment model entities at a given level. For the default ISA-95 template, the types shown here (process cell, production unit, production line, and storage zone) are included in the base system.

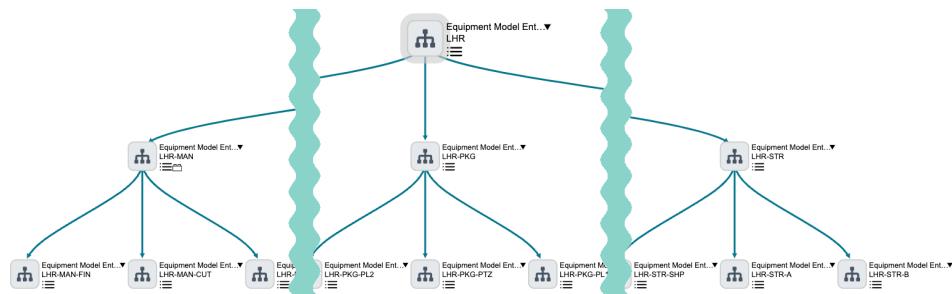


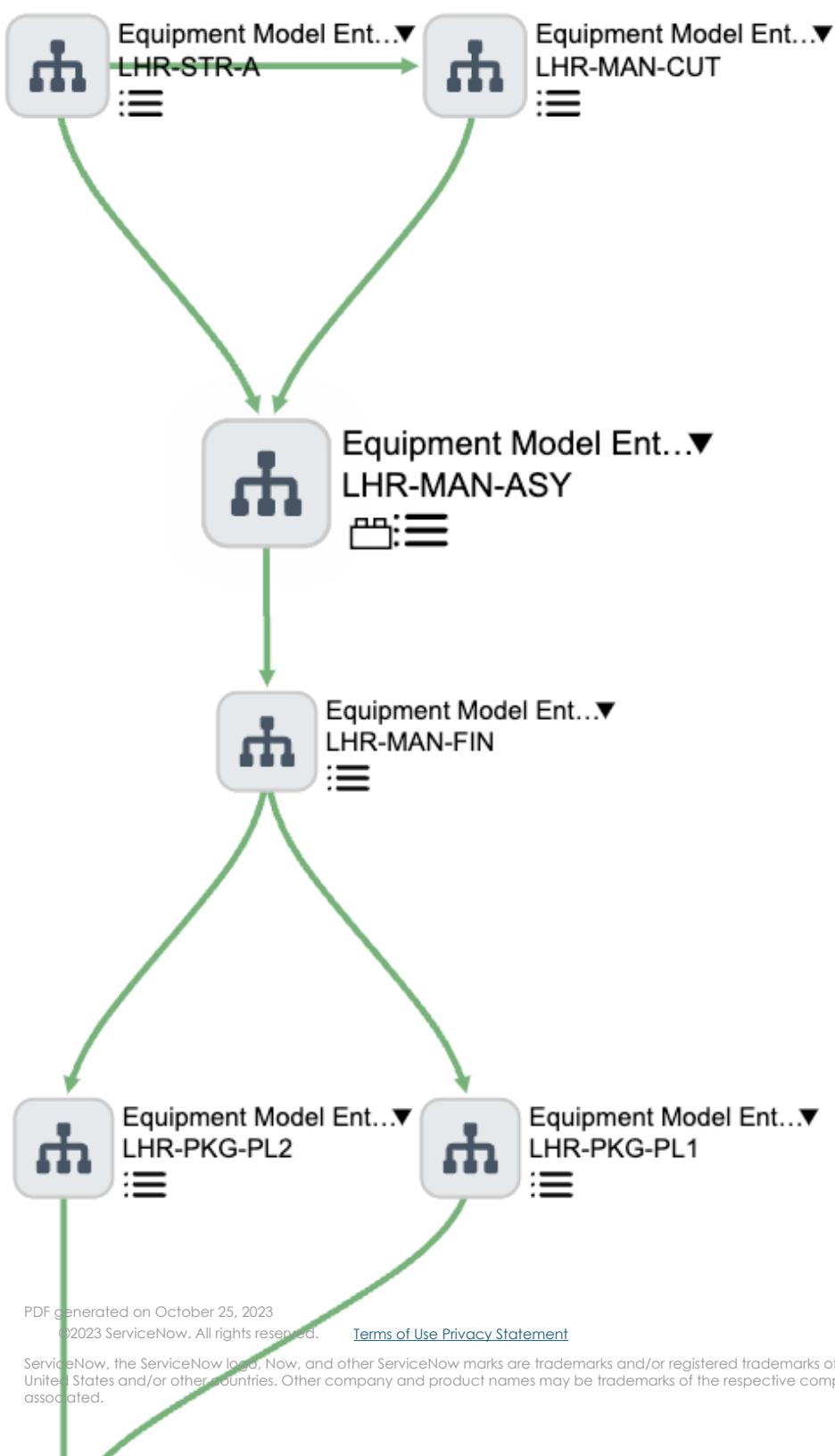
CSDM includes relationship types (specific to Operational Technology) that more accurately distinguish how OT assets and equipment model entities relate to each other:

- **Producer for::Consumer of:** Describes the production process (material flow) between equipment model entities.
- **Contains Element::Element of:** Describes the hierarchical relationship between equipment model entities.
- **Automated by::Automates:** Describes the relationship between an OT asset and an equipment model entity that the OT asset automates.
- **Detects::Detected by:** Describes which Network Intrusion Detection System (NIDS) class (`cmdb_ci_nids`) detected an OT asset on an OT network.

- **Owes::Owned by:** Describes the relationship when an OT Control Module is owned by an OT Control System (PLC, DCS, and so on)

The following dependency maps show the relationships between OT assets and equipment model entities:





Products that add value to Operational Technology

When you use OT with any of the following ServiceNow products, you increase the value you get from OT.

Discovery for Operational Technology

Discovery for Operational Technology provides details about IT-classed hardware and software CLs and can be configured to provide additional OT asset context like Purdue Level and Site on a per-OT schedule basis. Discovery for OT is part of the Operational Technology Manager product.

Industrial Process Manager

When OT assets are assigned to an equipment model entity, automated by::automates relationships are created between them. This can be done manually in the Industrial Workspace or using the relationship between OT subnets and equipment model entities using the Automatic Mapping Across Zone-based IP Network Groups (AMAZING) feature in the OT Subnet Mapping menu item.

Operational Technology Vulnerability Response

When vulnerable item (VI) records are created by importing records from an OT-certified integration with a third-party security platform, OT assets are associated with the VI. This enables both of the following capabilities:

- Risk calculation based on the criticality of the mapped equipment model entity.
- Assignment of VIs to the appropriate local team for remediation via site-based assignment groups.

Operational Technology Incident Management

Incident Management for OT runs separately from IT for most OT assets. OT incident records enable site-based access and views to issues that are related to OT assets.

Products that benefit from Operational Technology

IT Service Management (ITSM)

Services have the context of the site, production process, and OT assets, along with the information and technologies layered beneath them.

Information Technology Operations Management (ITOM)

Understands the business context for the production processes along with the OT asset hardware and software being managed.

Security Operations

Understands the business context for the production processes as well as OT asset hardware and software being secured.

Governance, Risk, and Compliance (GRC)

Auditors can better leverage production process flows and related Information objects. This helps auditors understand the design-time data sensitivity for scoping audits, measuring risks, and managing audit activities.

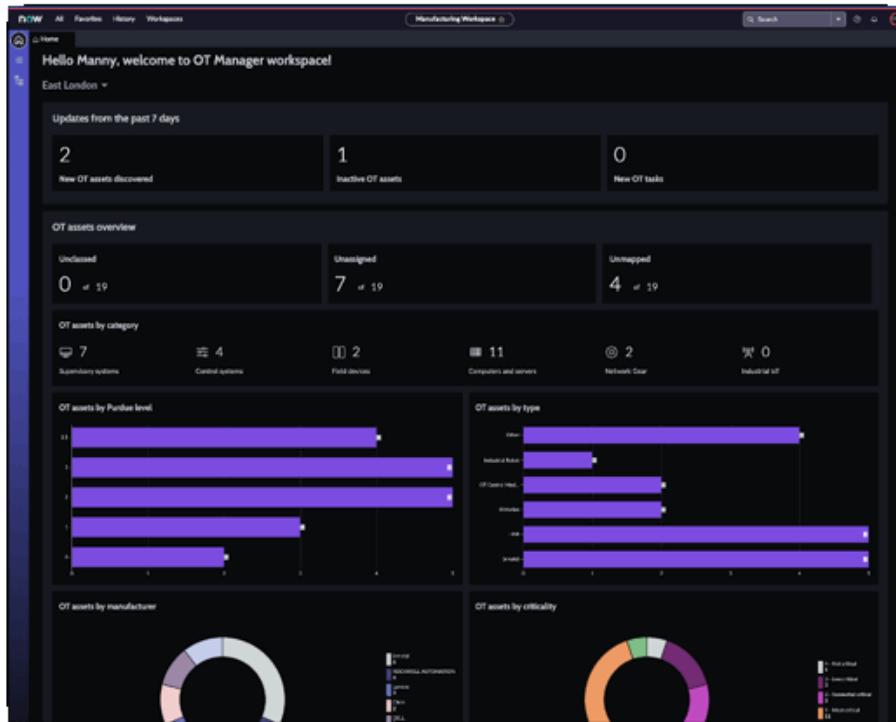
Asset Management

Manages the impact of the software and hardware life cycle process on the production processes.

The Operational Technology Manager use cases are described in this section.

Operational Technology Manager use case

The Operational Technology Manager application creates the foundational data and relationships that enable your enterprise to use the Operational Technology solution. It supports the use of the Configuration Management Database (CMDB), Service Graph connectors, and Discovery applications in the Now Platform.



The Operational Technology Manager workspace summarizes the OT asset inventory by Purdue Level and OT asset type.

Key features of the Operational Technology Manager use case

- Operational Technology Manager uses OT configuration item (CI) extension classes that extend the CMDB class hierarchy.
- Discovery for OT is restricted to OT specific roles and OT related meta data can be added to an OT discovery schedule.
- The Service Graph connector (Excel) imports OT data from a populated Microsoft Excel flat-file spreadsheet. This data is validated and then transformed into the appropriate table records and relationships to represent OT asset details in the CMDB.

Results of the Operational Technology Manager use case

With the Operational Technology Manager use case, you can:

- Visualize dependencies of OT assets in the industrial environment.
- Manage the life cycle of OT assets on a site-by-site basis with site specific RBAC
- Create a solid data foundation and define critical levels of infrastructure.

Industrial Process Manager use case

Use the Industrial Process Manager application to create the ISA-95 equipment model data foundation that is required for the Operational Technology solution.

The screenshot shows the ServiceNow interface for the Industrial Process Manager. The top navigation bar includes 'now', a search bar, and user icons. The main title is 'Equipment model' with a dropdown for 'Atlanta Site'. A 'Create new entity' button is visible. On the left, a sidebar shows a tree view of the Atlanta Site structure, including Building 22, Building 42, Model M and Q, Building 64, Model M, Model M, Work Cell 38, Work Cell 45, Model Q, Cell1, Cell2, and Model Q. The main content area is titled 'Building 64' and shows tabs for Details, Upstream Process, Downstream Process, Child Entities (4), and OT Assets. The 'Child Entities' tab is selected, displaying a table with four rows of data. The table columns are Entity name, Short code, Path, Template, Level, Type, Process criticality, Assigned to, Support group, and Managed By Group. The data is as follows:

Entity name	Short code	Path	Template	Level	Type	Process criticality	Assigned to	Support group	Managed By Group
Model M	MASS	ATL-B64-MASS	Atlanta Site Model	Production Area	Assembly Line	1 - most critical	James Smith	Atlanta Site Support	Atlanta Site Support
Model Q	QASS	ATL-B64-QASS	Atlanta Site Model	Production Area	Assembly Line	1 - most critical	James Smith	Atlanta Site Support	Atlanta Site Support
Model M	MPROD	ATL-B64-MPROD	Atlanta Site Model	Production Area	Production Line	1 - most critical	James Smith	Atlanta Site Support	Atlanta Site Support
Model Q	QPROD	ATL-B64-QPROD	Atlanta Site Model	Production Area	Production Line	1 - most critical	James Smith	Atlanta Site Support	Atlanta Site Support

Key features of the Industrial Process Manager use case

You can perform the following operations in the Industrial Process Manager workspace:

- Use ISA-95 models to describe the production process at each site in the industrial environment.
- Manage equipment model entities and their relationships with each other and with OT assets.

- Automatically map the relationship of all OT assets to the equipment model entity it automates using the OT subnet to equipment model entity relationship.

Results of the Industrial Process Manager use case

The Industrial Process Manager enables you to create a custom version of the equipment models in each of your sites.

Operational Technology Vulnerability Response use case

Operational Technology Vulnerability Response enables you to effectively prioritize and remediate OT asset vulnerabilities at the site level.

Key features of the Operational Technology Vulnerability Response use case

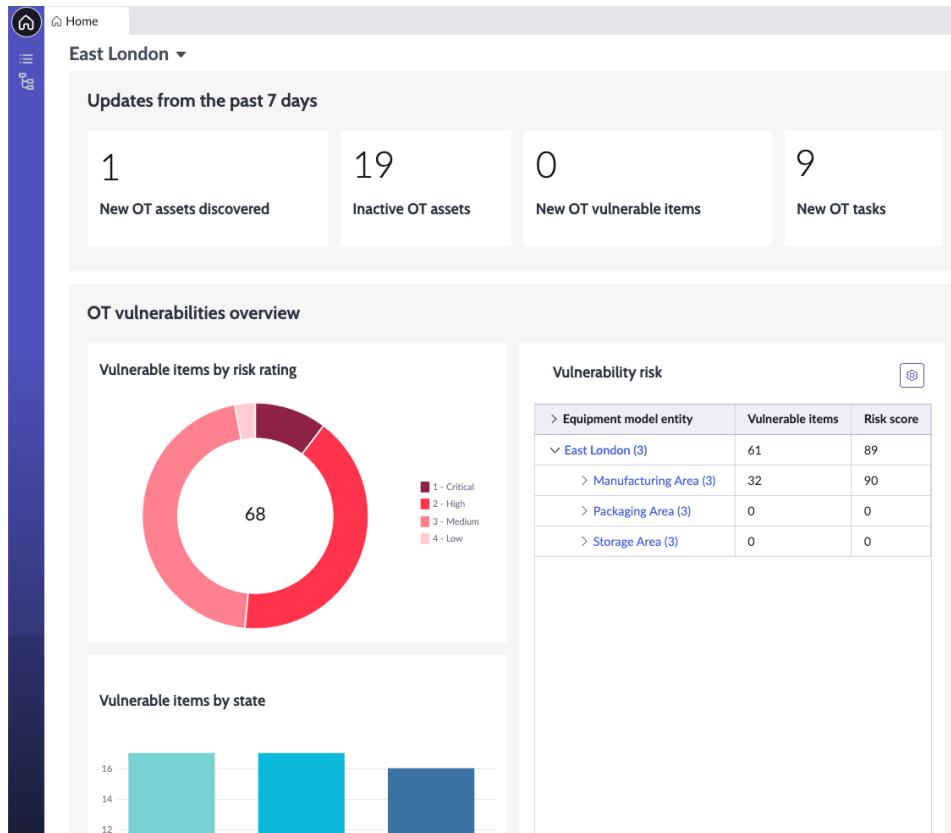
- Remediation owner workspace maps vulnerable items (VIs) with the production process.
- Risk calculation for OT VIs can be based on criticality of the equipment model entity automated by the OT asset.
- Automatic assignment of VIs to remediation owners based on the site assigned to the OT asset with the VI.

Results of the Operational Technology Vulnerability Response use case

By leveraging the CMDB relationships of OT assets, you can prioritize vulnerable assets or items based on the criticality of the production process they automate.

As an OT engineer or OT vulnerability manager, Operational Technology Vulnerability Response enables you to find answers to questions such as:

- What are my OT asset vulnerabilities?
- How can I prioritize vulnerability remediation using OT specific risk?
- What progress are we making toward remediation of OT vulnerabilities?



Operational Technology Vulnerability Response summarizes and rolls up vulnerability risk by equipment model entity.

Operational Technology Incident Management use case

OT incidents occur when there is a disruption in service that is provided by an OT asset on an OT network. Sometimes, the OT asset may not be known when the incident is first created. Operational Technology Incident Management enables engineers to quickly resolve OT asset and production process issues.

Key features of the Operational Technology Incident Management use case

When a user creates an OT incident from the Industrial Workspace, the incident is automatically assigned a Network Type of OT to distinguish an OT incident from an IT incident. The field is not displayed by default.

Results of the Operational Technology Incident Management use case

Operational Technology Incident Management enables engineers to quickly resolve OT asset and production process issues. It enables you to manage OT incidents separately from IT incidents.

Terms related to managing business applications with elements of CSDM.

Term	Definition
Equipment model	The service records that describe how an industrial operation is organized to produce an output or product.
Production process	The relationships between equipment model entities and the various stages of the workflow from the raw material to finished goods.
Site	A parent equipment model entity record that has no parent. This is a special equipment model entity record because it is used to assign read or write level access to the OT assets assigned to the site.
OT asset (site assignment)	The site assignment is needed for role-based security (RBAC) of OT assets. This is implemented as a choice list reference field on the OT Asset Details (cmdb_ot_entity) table portion of the OT asset record because an OT asset can belong to only one site.
OT asset (automates::automated)	The automates::automated by relationship describes how the OT asset is related to the production process, which could include more than one equipment model entity.

Term	Definition
Windows server (OT and IT networks)	In both OT and IT networks, the Windows server is represented in the cmdb_ci_win_server server. Additionally, the Windows server in the OT network has a reference in the cmdb_ci_win_server.cmdb_ot_entity field pointing to a record in the cmdb_ot_entity table that describes its function in OT and other OT characteristics like Purdue Level, site, and so on.

You might have questions while implementing the CSDM framework.

What is the difference between a production process and an equipment model entity?

Equipment model entities are the service records used to describe how an industrial operation is organized to produce an output or product. The production process describes the relationships between equipment model entities as material flows from raw input to a finished product.

What is a site?

A site is a parent equipment model entity record that itself has no parent. A site is a special equipment model entity record because it is used to assign read or write level access to the OT assets assigned to the site.

Why does an OT asset have both a site assignment and automates::automated by relationships?

The site assignment is needed for role-based security (RBAC) of the OT assets. This is implemented as a choice list reference field on the OT Asset Details (cmdb_ot_entity) table portion of the OT asset record because an OT asset can belong to one site only.

The automates::automated by relationship describes how the OT asset is related to the production process, which could include more than one equipment model entity.

What is the difference between a Windows server found on an OT network and one found on an IT network?

In both types of network, the Windows server is represented in the cmdb_ci_win_server server. Additionally, the Windows server in the OT network has a reference in the cmdb_ci_win_server.cmdb_ot_entity field pointing to a record in the cmdb_ot_entity table that describes its function in OT and other OT characteristics like Purdue Level, site, and so on.

ITOM Health product view

ITOM Health includes the Event Management and ITOM Health applications, which help you track and maintain the health of services in your organization. The goal of this product view is to help you to understand how ITOM Health key entities work with the core CSDM framework.

ITOM Health consists of two main capabilities that use the CSDM framework.

Event Management

Event Management gathers alerts from infrastructure events captured by third-party monitoring tools. Event Management uses IT-related information gathered by Discovery to map alerts to Cls. Based on the collected information, Event Management then provides dashboards showing a consolidated view of all service-impact events.

Metric Anomaly Detection

Proactively analyzes your IT infrastructure to spot issues and prevent service outages. Using advanced machine learning to analyze information about your IT infrastructure, the application automatically determines dynamic thresholds and identifies anomalies that may indicate potential service outages.

For more information on ITOM Health, see [ITOM Health](#).

- [ITOM Health and CSDM tables](#)

ITOM Health manages and uses CSDM tables. Several ServiceNow products benefit from and add value to ITOM Health.

- **ITOM Health use case**

Modern organizations increasingly rely on artificial intelligence (AI) technologies in IT operations (AIOps) to help address rapid growth in data volumes and variety. Modern organizations need to analyze this data and find ways to automate and predict issues before they occur. AIOps platforms have emerged as a solution to many of these challenges.

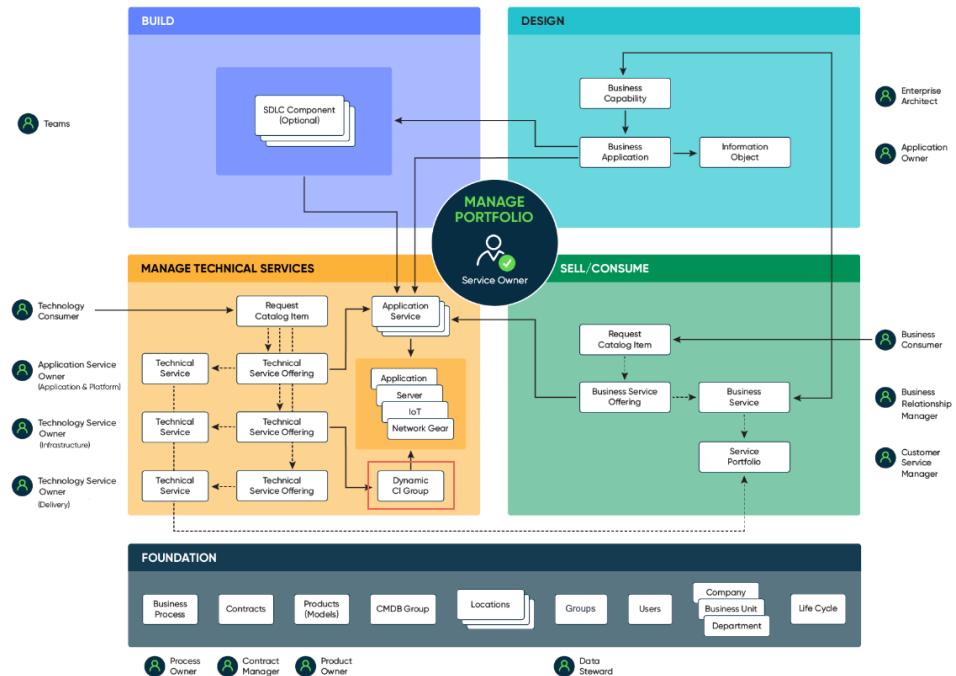
- **ITOM Health considerations**

Consider these points while implementing the CSDM framework.

ITOM Health manages and uses CSDM tables. Several ServiceNow products benefit from and add value to ITOM Health.

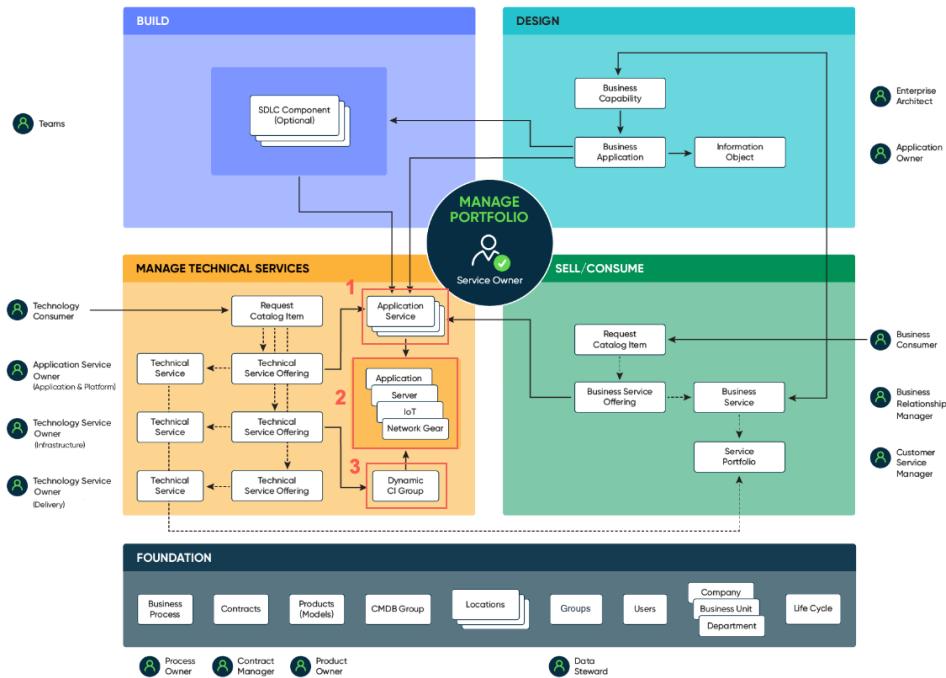
CSDM table managed by ITOM Health

ITOM Health manages the Dynamic CI Group table
[cmdb_ci_query_based_service]



CSDM tables used by ITOM Health

1. Mapped Application Service table [cmdb_ci_service_discovered]
2. Configuration Item table [ci_*]
3. Dynamic CI Group table [cmdb_ci_query_based_service]



Products that add value to ITOM Health

When you use ITOM Health with other ServiceNow products, you increase the value you get from ITOM Health. These other ServiceNow products include:

Discovery

Discovery provides details about the hardware and software CIs you are using.

Service Mapping

Service Mapping provides details about the application instance service in the [cmdb_ci_service_discovered] table, relating infrastructure and application [cmdb_ci_appl] CIs.

Service Portfolio Management (Service Portfolio Management)

Provides the related product model. Software Asset Management (SAM) and Hardware Asset Management (HAM) provide life-cycle data for Technology Portfolio Management (TPM).

Products that benefit from ITOM Health

Incident Management

Incidents are created using the downstream information from Event Management.

Customer Service Management (CSM)

ITOM Health provides the application service impact which, in turn, is used to identify the affected users.

Modern organizations increasingly rely on artificial intelligence (AI) technologies in IT operations (AIOps) to help address rapid growth in data volumes and variety. Modern organizations need to analyze this data and find ways to automate and predict issues before they occur. AIOps platforms have emerged as a solution to many of these challenges.

Key features of the ITOM Health use case

- Event and metrics ingestion using the base-system integration to selected monitoring systems
- Event and metric processing and alert creation
- Alert correlation and root cause analysis
- Anomaly detection
- Application service impact calculation
- Service Operations workspace

- Alert remediation automation

	Ingestion 	Processing 	Correlation, RCA & Anomaly Detection 	Service Impact 	Visualization 	Automation
Events	<ul style="list-style-type: none"> • OOB connectors • 3rd-party connectors* • Web services • SNMP traps • Email • > 1600 events / sec 	<ul style="list-style-type: none"> • De-duplication • Filtering • Transform • Threshold • Bind to CIs 	<ul style="list-style-type: none"> • Correlation rules • ML-based grouping • CMDB-based grouping • Root cause analysis • Correlate detected CI & service topology changes 	<ul style="list-style-type: none"> • Identify services impacted by alerts • Service health topology maps • Calculate alert priority • Blackout support on alerts and service impact 	<ul style="list-style-type: none"> • Operator Workspace for service health • Alert intelligence for alert management and response • Mobile application for iOS and Android 	<ul style="list-style-type: none"> • Auto-open incidents & other tasks • Trigger <ul style="list-style-type: none"> - Workflows - Orchestrated diagnostics and remediations
Metrics	<ul style="list-style-type: none"> • OOB connectors • 3rd-party connectors* • Web services • Ingestion rate: 800k metrics / min 	<ul style="list-style-type: none"> • Auto-register metric types • Map metrics to CIs • Classify and model metric behavior • Calculate upper/lower control bounds 	<ul style="list-style-type: none"> • Compare raw metrics to projected bounds • Identify and score anomalous metrics • Create anomaly alerts for "golden" metrics 	<ul style="list-style-type: none"> • Relate CI metrics to services 	<ul style="list-style-type: none"> • Anomaly Map • Insights Explorer (IE) • Advanced Insights Explorer • Metrics in Alert Intelligence 	<ul style="list-style-type: none"> • Promote anomaly alerts to IT alerts for processing in Event Management • Anomaly alert CI blacklisting

Results of the ITOM Health use case

Application service maps help Network Operations Center (NOC) operators at central locations to accurately predict root cause alerts and correlate the alerts with the discovered topology data. Correlating the alerts with the discovered topology data lets you accurately predict the impact to services. Impact analysis helps you to determine the impact on the application services and visualize it in the application service map.

Consider these points while implementing the CSDM framework.

Considerations for implementing ITOM Health

CIs required to get the most benefit from Event Management

Event Management uses the CIs that are monitored by the operations team to bind an alert to the correct CI and analyze the impact on the monitored service.

Technical Services in Event Management

Technical Services are now labeled as Dynamic CI Group service. Dynamic CI Group services use the CMDB CI Group capability to query the CMDB.

Common use case for Dynamic CI Group service

The Dynamic CI Group service is used in Event Management as a logical grouping of CIs. Dynamic CI Group service provides the health status of the group to the technology or service owner.

ITOM Visibility product view

ITOM Visibility consists of two ServiceNow products: Discovery and Service Mapping. These products are responsible for creating Configuration Items (CIs) in the CMDB and relating them. The goal of this product view is to help you to understand how ITOM Visibility works with the core CSDM framework.

Aspects of ITOM Visibility

Discovery

Discovery finds computers, servers, printers, a variety of IP-enabled devices, and the applications that run on them. It can then update the CIs in your CMDB with the collected data.

Service Mapping

Service Mapping discovers all application services in your organization and builds a comprehensive map of all devices, applications, and configuration profiles that support the associated business services.

Service Mapping maps dependencies based on the connections between devices and applications. This method, called top-down mapping, helps you immediately see the impact of a problematic object on the rest of the application services.

For more information

[Discovery basics](#).

[Service Mapping](#).

- [ITOM Visibility and CMDB tables](#)

ITOM Visibility manages and uses CMDB tables. Several ServiceNow products benefit from and add value to ITOM Visibility.

- **ITOM Visibility use case**

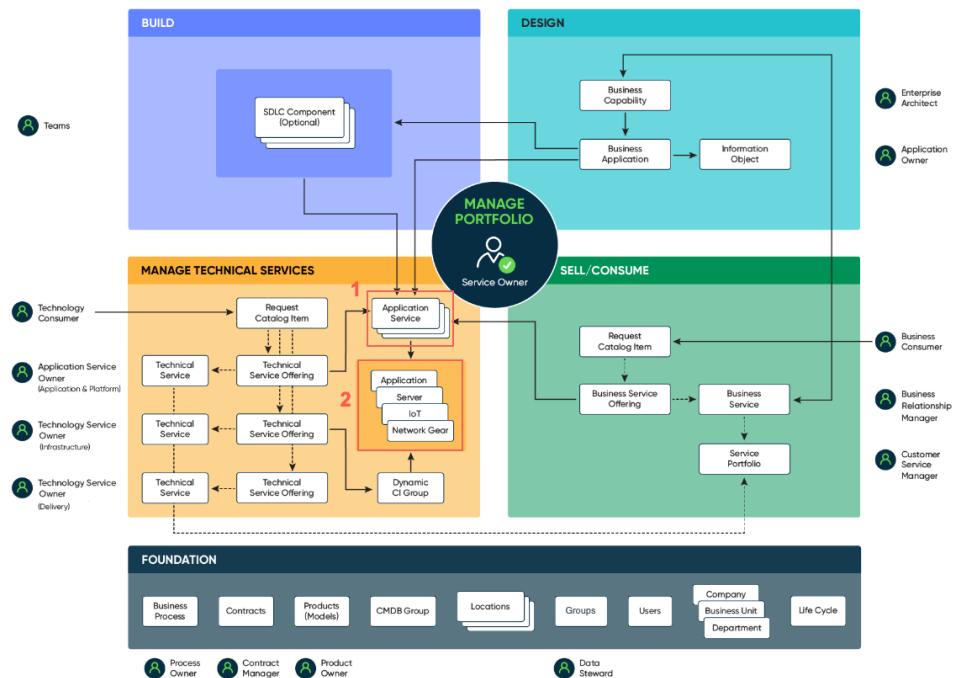
The ITOM Visibility use cases are described in this section.

- **ITOM Visibility considerations**

Consider these points while implementing the CSDM framework.

ITOM Visibility manages and uses CMDB tables. Several ServiceNow products benefit from and add value to ITOM Visibility.

Tables that ITOM Visibility manages



Mapped application service tables

The tables that ITOM Visibility manages depend on the method used to map the application service. You can use the Application Service wizard to populate an application service.

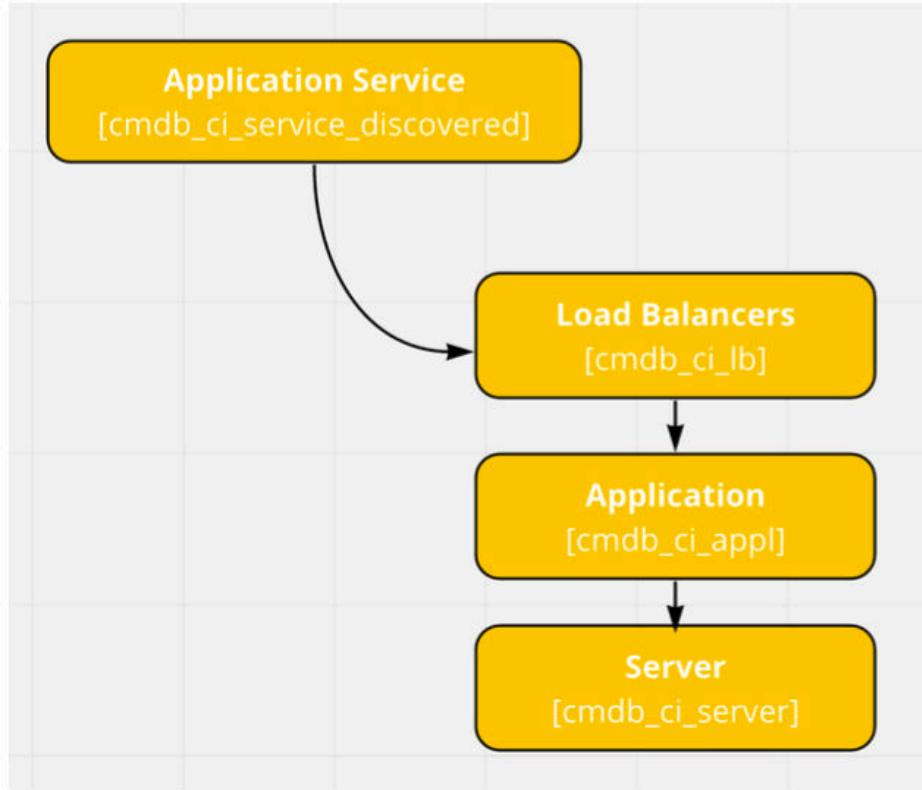
- Map the application service using top-down discovery of CI connections. This method requires Service Mapping, the sm_admin

role, and the current domain must be a leaf domain. This is the best option for accuracy and complete mapping. ITOM Visibility manages the Mapped Application Service table [cmdb_ci_service_discovered]

- Map the application service using a tag list or tag family. This method requires Service Mapping and is preferred for public or private cloud where tags are typically used. ITOM Visibility manages the tag-based maps table [cmdb_ci_service_by_tags].
- Manually map the application service. This method requires that you select specific CIs — no automation is used. ITOM Visibility manages the service maps table [cmdb_ci_service_manual].
- Use Dynamic Service to synchronize manually-created CI relationships to a Service Map view.

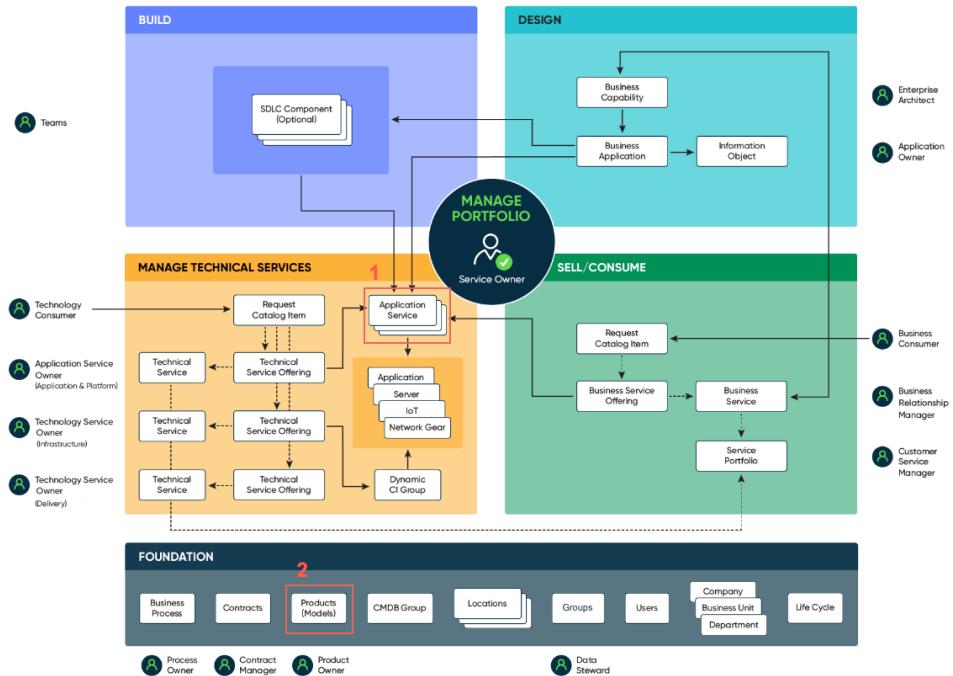
Configuration items (CIs) tables [cmdb_ci_*]

- Application table [cmdb_ci_appl]
- Server table [cmdb_ci_server]
- Virtual machines table [cmdb_ci_vm_instance]
- Load balancer table [cmdb_ci_lb]
- Network gear table [cmdb_ci_netgear]



Tables that ITOM Visibility uses

1. Dynamic CI Group table [cmdb_ci_query_based_service]
2. Product model tables (for Technology Portfolio Management [TPM])
software and hardware models



Products that benefit from ITOM Visibility

- IT Service Management
- Customer Service Management (CSM)
- Event Management
- Cloud Management
- Asset Management (Hardware Asset Management and Software Asset Management)
- Strategic Portfolio Management (Financial Management — Showback statements, Application Portfolio Management [APM])
- Security Operations (Incident Response, software vulnerability management)

The ITOM Visibility use cases are described in this section.

Discovery use case

- Gain increased visibility into your on-premises and cloud resources.
- Keep track of changes in your on-premises, cloud, and serverless infrastructure in the Configuration Management Database (CMDB).
- Set a strong foundation with accurate data and relationship views for ServiceNow products, including Change Management, Software Asset Management, Customer Service Management (CSM), and Security Operations.

Key features of the Discovery use case

- Set up and manage Discovery jobs.
- See IT resources and dependencies at-a-glance.
- Build queries to validate discovered IT resources.
- Manage your public key infrastructure (PKI) certificates in one dashboard.

Results of the Discovery use case

The CSDM framework gives Discovery a prescribed model for how infrastructure and software CIs relate to other areas, such as application services, business applications, and assets.

Service Mapping use case

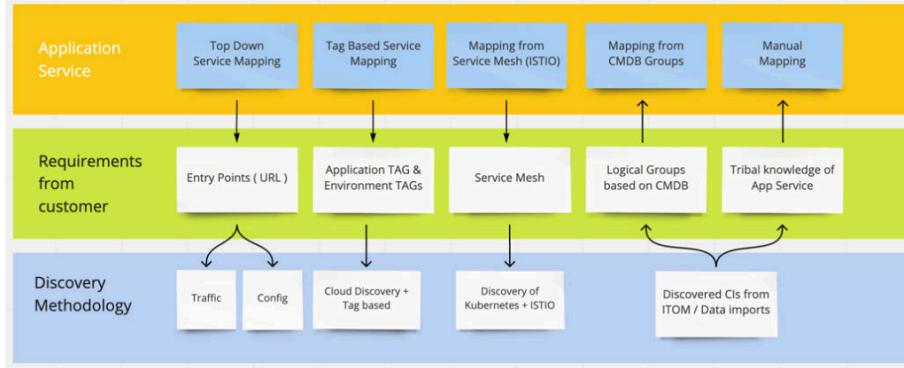
You can create application services manually, using an API, or by having Service Mapping discover them. Regardless of creation method, all application services are stored in the Mapped Application Service table [cmdb_ci_service_discovered].

Results of the Service Mapping use case

Service Mapping automates a critical aspect of CSDM with a consistent, automated approach to connect the logical layer of the CSDM model to the physical model CIs in the CMDB. This approach lets you more effectively manage your business applications. The approach also enables you to automate modeling of your business applications for impact assessment and analysis. For impact assessments and analysis,

you can use a number of ServiceNow products including Change Management, Incident Management, or the CMDB Query Builder.

Application Services flow



Application services view in Operator Workspace

You can specify a life-cycle status for an application service. Application services with an Operational life-cycle status can be used on service maps for the relevant workflows, such as Artificial Intelligence for IT Operations (AIOps) workflows.

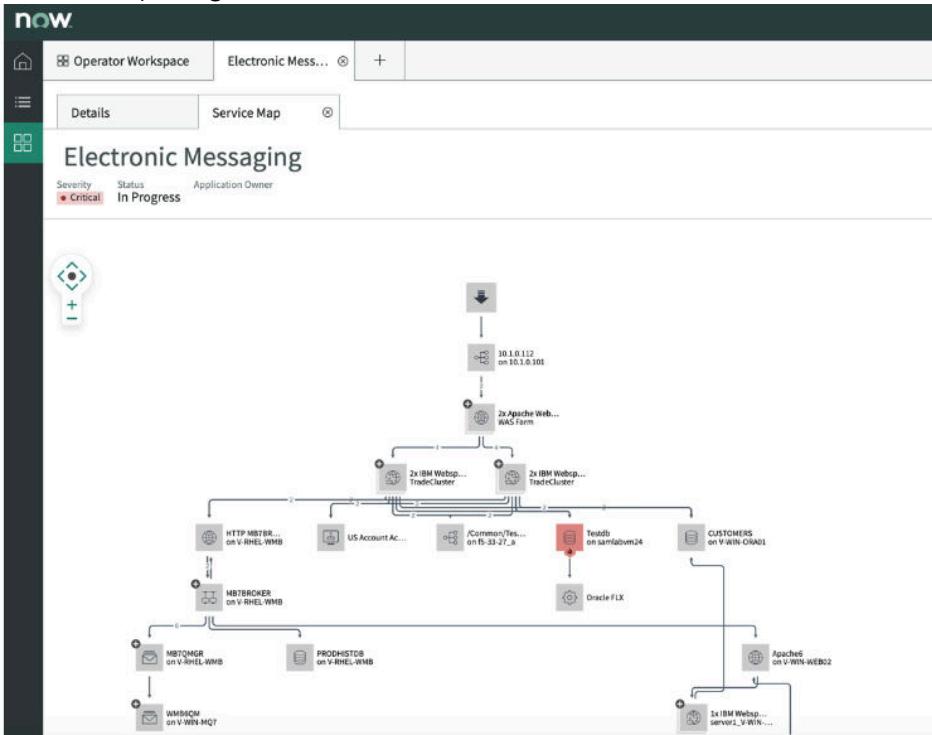
- ✓ **Operational**
- Non-Operational**
- Repair in Progress**
- DR Standby**
- Ready**
- Retired**

You can view information about the services, such as the criticality, in the Operator Workspace.

Operator Workspace

The screenshot shows the ServiceNow Operator Workspace interface. At the top, there's a navigation bar with tabs like Home, Operator Workspace, and a search bar. Below the navigation is a summary section for 'All Services (30)' with metrics: Critical (0.6%), Major (86.7%), Minor (5.1%), Warning (0.0%), and Info (25.8%). A 'Filter by Alert' sidebar on the right displays two alerts: 'CPU percentage over 80 percent' and 'Anomaly score 9.184539'. The main area is titled 'Operator Workspace' and contains a 'Service Map' diagram. The map is organized into three categories based on business criticality: 'most critical' (red), 'somewhat critical' (orange), and 'less critical' (yellow). Each category contains several service nodes, each with a small icon and a label. For example, under 'most critical', there are nodes for 'Electronic Messaging' (red), 'QA Audit' (orange), 'Production Audit' (orange), and 'EU - Customer Purchase' (orange). Under 'somewhat critical', there are nodes for 'Loyalty Club' (orange), 'DevOps Test Management' (orange), 'DevOps Source Code M...' (orange), 'UK Portal' (green), 'Supply Chain Manag...' (green), 'Jenkins' (green), and 'EMEA Portal' (green). Under 'less critical', there are nodes for 'Demographics Research' (orange), 'User Account Access' (orange), 'Credit Check' (orange), 'Purchasing Trend...' (orange), 'APAC Account Access' (orange), 'Billing' (orange), 'User Verification' (orange), 'CRM' (orange), 'Consumer Analytics' (orange), 'Customer Purchases' (orange), 'Asia Portal' (yellow), 'Customer Loyalty' (green), and 'Demand Mkt' (green).

The Service Map tab displays a visual representation of the service.
Service map diagram



Related concepts

- [Document life cycle](#)
- [Hardware life cycle](#)
- [Location life cycle](#)
- [Logical life cycle](#)
- [Product life cycle](#)

Consider these points while implementing the CSDM framework.

You can use Service Mapping to define application services.

Here are alternative methods for defining application services:

- Define manually.
- Use a Dynamic CI group, based on a query.
- Use a tag-based approach (commonly used for Cloud integrations).

For more information about creating application services, see [Create an application service](#).

Problem Management product view

Problem Management helps identify the cause of an error in the IT infrastructure, reported as occurrences of related incidents. The goal of this product view is to help you to understand how Problem Management key entities work with the core CSDM framework.

With Problem Management you can perform the following actions:

- Document the root cause of an error and resolve it permanently.
- Create known error articles from a problem to provide guidance and help deflect incidents.
- Collaborate on problem tasks with other teams to identify the root cause using a drag-and-drop functionality on visual task boards.

The Problem Management form references the following CSDM elements (attributes and related views).

1. Service — References the [cmdb_ci_service_business] table.
Note: Earlier releases labeled this attribute Business Service.
2. Service Offering — References the [service_offering] table to see the service offerings affected by the problem in the Service Offerings related list [task_service_offering].
3. Configuration Items — References the [cmdb_ci] table
4. Affected/Causal CIs — Related list [task_ci] table
5. Impacted Services — Related list [task_cmdb_ci_service] table

Change Request form

- Problem Management and CSDM tables

Problem Management manages and uses CSDM tables. Several ServiceNow products benefit from and add value to Problem Management.

- **Problem Management use case**

The Problem Management use case is described in this section.

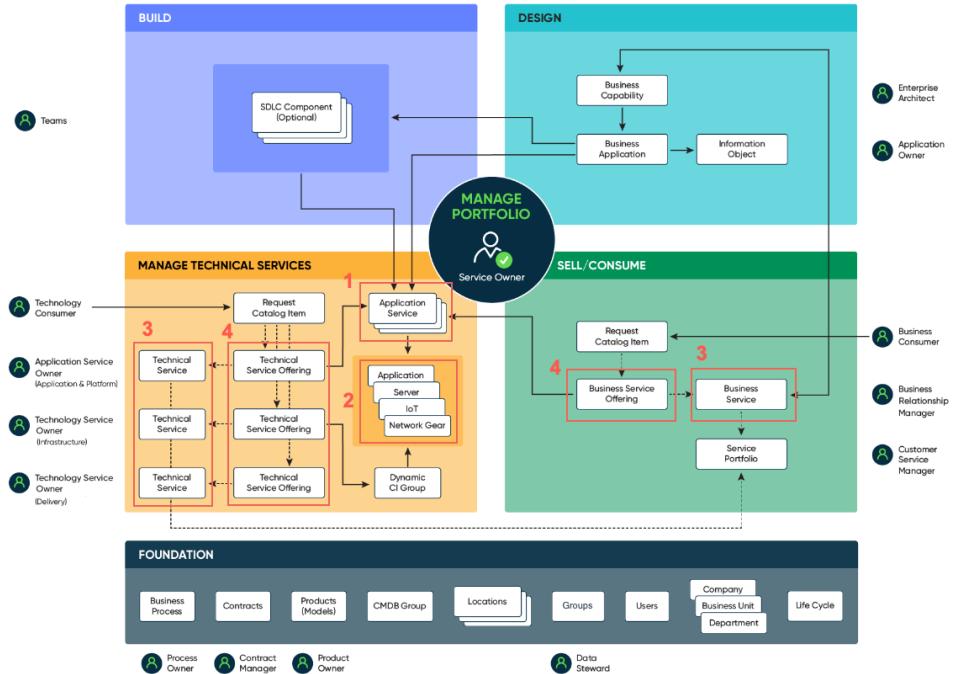
- **Problem Management considerations**

Consider these points while implementing the CSDM framework.

Problem Management manages and uses CSDM tables. Several ServiceNow products benefit from and add value to Problem Management.

CSDM tables used by Problem Management

1. Application Service table [cmdb_ci_discovered_service]: Application Service or any infrastructure CI
2. Configuration Item tables [cmdb_ci*]
3. Business Service [cmdb_ci_service_business] table and Technical service [cmdb_ci_service_technical] table: Use the service classification attribute to identify business services and technical services.
4. Service Offering table [service_offering]: Utilized as a choice list attribute to filter types of service offerings like Business Service, Technical Service, and Application Service.



Products that add value to Problem Management

When you use Problem Management with one of the following ServiceNow products, you increase the value you get from Problem Management.

- Discovery provides details about the hardware and software CIs you are using.
- Service Portfolio Management provides life-cycle information for a service.
- Asset Management provides the related product model. Software Asset Management (SAM Foundation) and Hardware Asset Management (HAM) provide life-cycle data for Technology Portfolio Management.
- Security Management provides initial information to containment, eradication, and recovery of security related problems.
- Risk Management provides IT risk and financial risk information.

Products that benefit from APM

- IT Service Management (ITSM): Services have the context of the business and applications, along with the information and technologies layered beneath them.
- Information Technology Operations Management (ITOM): Understands the business context for the application services along with the hardware and software being managed.
- Governance, Risk, and Compliance (GRC): Auditors can leverage the business applications and related Information Objects. This helps auditors understand the design-time data sensitivity for scoping audits, measuring risks, and managing audit activities.
- Asset Management: Manages the software and hardware life cycles for business applications and business services.

The Problem Management use case is described in this section.

Problem Management is used to prevent problems and the occurrence of resulting incidents. It also aims at eliminating recurring incidents and minimizing the impact of incidents that cannot be prevented. With Problem Management, you can capture information on affected configuration items (CIs), with type as asset, in a problem to keep a record of the updated, repaired, swapped, or retired configuration items. By keeping track of the assets, you can identify the location of the assets, their usage and when the assets were changed. Using Problem Management, you can monitor and manage the assets in your company using a systematic approach.

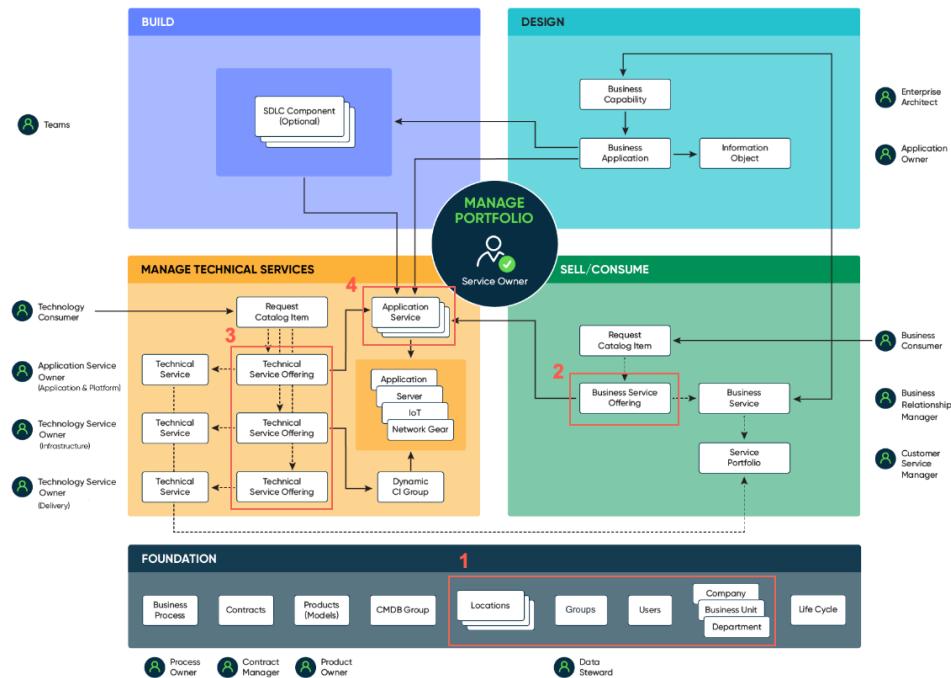
If a configuration item (CI) has resulted in a problem, use the dependency view to identify other configuration items (CIs) affected by the CI that caused the problem. You can then associate affected configuration items (CIs) with a problem record to find out how the problem affects other CIs with dependent relationships.

Key features of the Problem Management use case

The CMDB, when used by the CSDM framework, provides value to Problem Management in the following ways:

- Understand the impact of the problem on services and service offerings.

- Dynamically route problems.
- Identify one or more impacted services to address the problem.



The CSDM data elements used in Problem Management are:

1. Subscription: Related lists on service offerings that identify who has access to the offering and thus may be impacted in an outage. A problem can identify impact using the subscribed by tables. The related lists are as follows:
 - Service Subscriptions by Company [service_subscribe_company]
 - Service Subscriptions by Department [service_subscribe_department]
 - Service Subscriptions by Group [service_subscribe_sys_user_grp]
 - Service Subscriptions by Location [service_subscribe_location]
 - Service Subscriptions by User [service_subscribe_sys_user]

2. Business service offering may be used by problems to provide the business approver based on approval_group and business_criticality. A business service may have multiple offerings, each with a different criticality.
3. Technical service offering may be used by problems to provide the technical approver approval_group and technical assignment group on the attribute assignment_group.
4. Application service may be used to provide prod and non-prod (DEV, QA, UAT, etc.) environments. Non-prod environments may be filtered out if desired. The legacy **used_for** attribute maps to the **environment** attribute. You should use the **environment** attribute.

Note: Some service offerings may identify the environment of the offering as well.

Results of the Problem Management use case

The CSDM framework provides Problem Management context for problems on what CIs may be involved.

To determine the impact and root cause, complete the following steps:

1. Populate the Configuration Item attribute on the Problem form, configuration_item, with the CI item or service affected.
2. (Optional) Use the Service and Service Offering attributes on the Problem form to help narrow down the list of configuration items to choose from. This feature is not available with the base system and needs additional configuration.
3. (Optional) Use the Affected CI related list to identify the CIs that may have caused the problem.

Consider these points while implementing the CSDM framework.

- Application Service: An application service is a service type that is a logical representation of a deployed application stack.
- Using application service for a problem: Some of the scenarios where the application service can be used for a problem include:
 - The problem is for an application issue: In this scenario the application service may be populated in the configuration_item attribute on the

Problem form representing the unique deployment of an application stack. For example, the application service of MyApp 3.0 Prod has been reported as being unavailable.

- The problem on an infrastructure CI is impacting devices. In this scenario the application service may be populated on the problem's Impacted Services related list, task_cmdb_ci_service, to identify the Application Service as one of the impacted services. For example, the Server Acme42 may be impacting my Application Service of MyApp 3.0 Prod and other related services.

Service Catalog product view

The Service Catalog lets you create other catalogs (such as the Request Catalog and the Product Catalog) that provide self-service opportunities in the channel you want to use, such as the self-service portal, the mobile app, or the Virtual Agent (conventional interface). The goal of this product view is to help you to understand how Service Catalog key entities work with the core CSDM framework.

You can create catalog items that can be used submit requests such as service and product offerings. You can also standardize request fulfillment to ensure the accuracy and availability of the items in the catalogs.

In addition to offerings that are fulfilled using Request Management application for IT, you can use Service Catalog Record Producers to submit requests or tickets in non-IT applications such as HR, Legal, Facilities and any record in any custom application.

- [Service Catalog and CSDM tables](#)

Service Catalog manages and uses CSDM tables. Several ServiceNow products benefit from and add value to Service Catalog.

- [Service Catalog use case](#)

You can create customized catalogs where you can request items such as a specific service or product. With this use case, CSDM provides Service Catalog connection to the service offerings and services.

- [Service Catalog considerations](#)

Consider these points while implementing the CSDM framework.

Service Catalog manages and uses CSDM tables. Several ServiceNow products benefit from and add value to Service Catalog.

CSDM tables managed by the Service Catalog

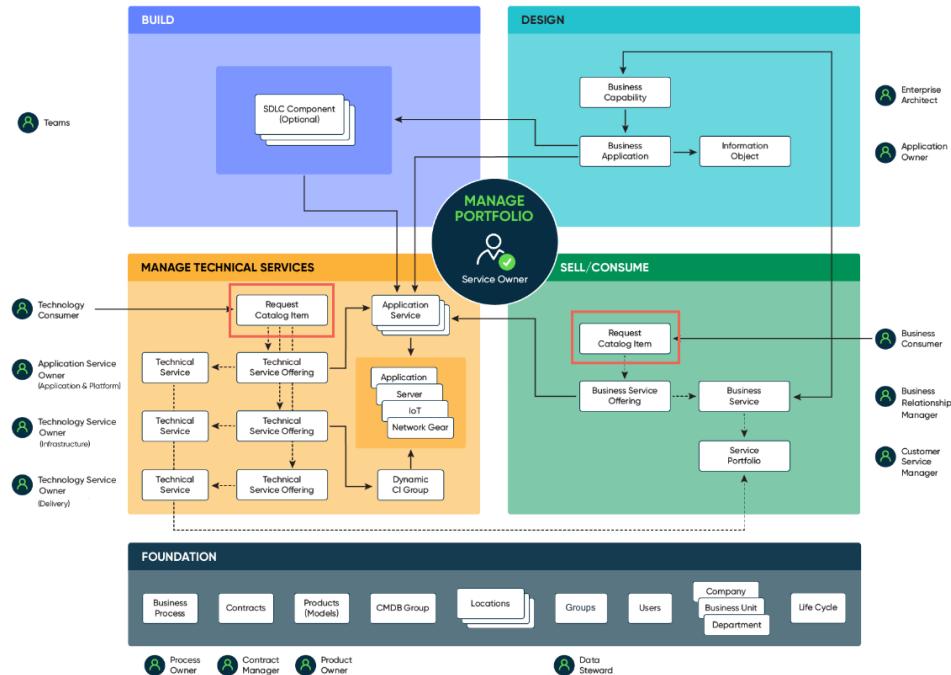
The Service Catalog primarily manages business services in the Sell/Consume domain. Technical Services are also viewed from the Operate domain of Event Management.

The Service Catalog manages the Catalog Item table [sc_cat_item]. The Catalog Item table creates requests in Request Management. The requests usually have an automated, semi-automated, or business workflow for fulfilling the request, which can consist of approvals and tasks. When part of Service Portfolio Management, the CIs are associated with service offerings. When you link a catalog item, you can track request activity for all catalog items associated with a service offering. An offering can have multiple catalog items.

The CIs can include:

- PC Hardware Item (pc_hardware_cat_item): Submits hardware asset requests included in Asset Management workflows.
- PC Software Item (pc_software_cat_item): Submits software asset requests included in Asset Management workflows.
- Record Producer (sc_cat_item_producer): Submits requests or generates records other than Request Management tables for services that aren't serviced by Request Management (for example, HR Cases, Facilities Requests, or Legal Requests).

CSDM tables managed by the Service Catalog



CSDM tables used by the Service Catalog

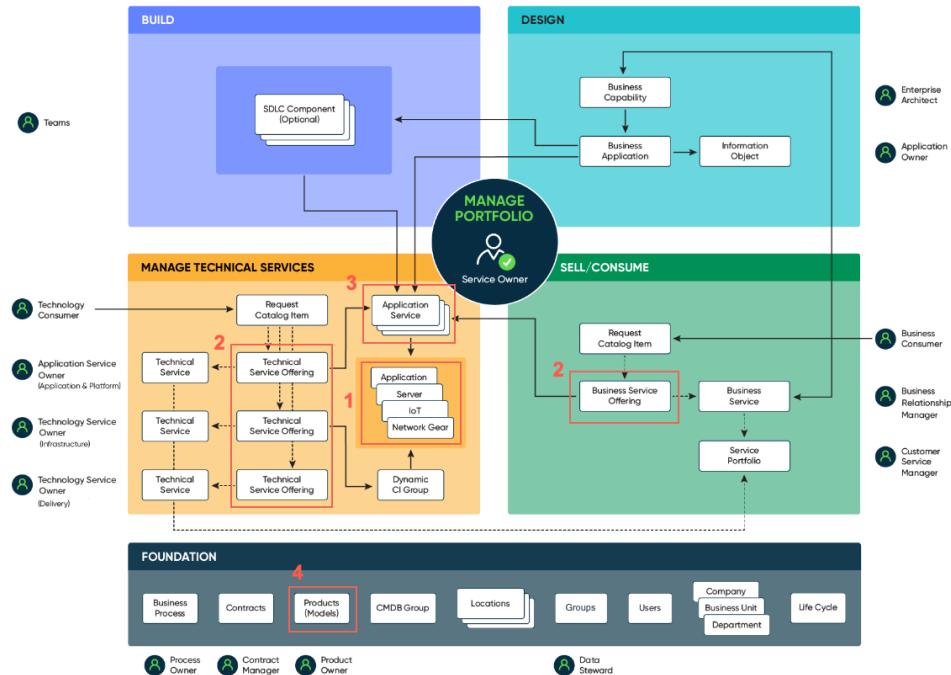
1. Configuration Item tables [cmdb_ci*]
2. Service Offering table [sc_cat_item_subscribe]
3. Application Service table [cmdb_ci_discovered_service]
4. Product Model tables [cmdb_model]

Note: The following legacy CMDB relationships that once provided access to catalog items are no longer supported. To provide access, use the new relationships listed in the table.

Relationship tables

CMDB table	Catalog item relationship table	New relationship to use instead
User (sys_user)	sc_cat_item_user_mt om, sc_cat_item_user_no _mtom	User criteria
Group (sys_user_group)	sc_cat_item_group_ mtom, sc_cat_item_group_n o_mtom	User criteria
Department (cmn_department)	sc_cat_item_dept_mt om, sc_cat_item_dept_no _mtom	User criteria
Location (smn_location)	sc_cat_item_location _mtom, sc_cat_item_location _no_mtom	User criteria
Company (core_company)	sc_cat_item_compa ny_mtom, sc_cat_item_compa ny_no_mtom	User criteria

CSDM tables used by the Service Catalog



Products that benefit from the Service Catalog

Service Portfolio Management (Service Portfolio Management)

Ties catalog items to service offerings. Lets service owners create and maintain catalog items more easily, and gives more visibility into the service offerings.

Hardware Asset Management and Software Asset Management

Self-service catalog lets you order an asset and track service delivery.

Human Resources (HR)

Exposes the creator (Record Producer) of HR cases and displays the Record Producer in the relevant self-service catalogs (for example, Portal, Mobile, and Virtual Agent).

Customer Service Management (CSM)

Exposes the creator (Record Producer) of customer service cases and displays the Record Producer in the relevant self-service catalogs (for example, Self-service Portal, Mobile, and Virtual Agent).

You can create customized catalogs where you can request items such as a specific service or product. With this use case, CSDM provides Service Catalog connection to the service offerings and services.

Service Catalog use case

Catalogs contain catalog items and are the starting points for accessing available services. Request Catalogs and Product Catalogs are two of the customized catalogs you can create from the Service Catalog.

-

A Request Catalog is a list of business and technical products, services, service commitment options, and offerings that you can order.

-

A Product Catalog is a set of information about individual models. A model is a specific version or configuration of an asset. Asset managers use the Product Catalog as a centralized repository for model information.

Key features of the Service Catalog use case

Digital Portfolio Management (DPM) uses the CSDM framework to navigate the Service Portfolio. You can then use the Service Portfolio to locate the services and their related service offerings, catalog items, dependents, dependencies, metric roll-ups, costs and initiatives. The associated metrics are aggregated using the CSDM framework with the related tables.

Catalog items

A catalog item is an item or a service that you can request from the catalog. A service can contain multiple catalog items. Catalog items are listed on the catalog portal and are available to the users who need them (either through subscription or job responsibility). Each catalog item is linked to one service offering.

The Catalog Item form includes the following tabs.

- Item Details: Description of the catalog item.
- Process Engine: The defined process used to fulfill the catalog Item request.
- Picture: Picture used to represent the catalog item when it appears in the catalog.
- Pricing: Pricing details for the catalog item.
- Portal Settings: Provides specific settings on how you interact with the catalog item in the catalog portal.

Catalog Item form

The screenshot shows the ServiceNow Catalog Item form for a 'Sales Laptop'. The top navigation bar includes icons for back, forward, search, and various actions like Update, Copy, Try It, and Delete. Below the bar, a message states: 'Catalog items are goods or services available to order from the service catalog. Items can be anything from hardware, like tablets and phones, to software applications, to furniture and office supplies.' It also lists validation rules: 'Enter a Name and Short description to display for the item.' and 'Enter a Price, approvals, variables, and other information as needed.' The main content area has tabs for Item Details, Process Engine, Picture, Pricing, and Portal Settings. The Item Details tab is active and highlighted with a red border. The content includes fields for Name ('Sales Laptop'), Application ('Global'), Catalogs ('Service Catalog'), and Active status ('Active'). Below these are sections for Short description (containing 'Corporate standard laptop for sales employees') and Description (containing rich text editor controls and a detailed description of the laptop). A 'Item Includes' section lists: '2.5 GHz processor', '750 GB Hard Drive', '8 GB RAM', 'Latest operating system', and 'Productivity software'. At the bottom, there's a note about adding relevant tags to the Meta field.

Product Catalog use case

Use the Product Catalog to specify information about a product model. Product models are specific versions or configurations of an asset. Asset managers use the Product Catalog as a centralized repository for model information.

A detailed and well-maintained Product Catalog can coordinate with the Service Catalog, asset, procurement, request, contract, and vendor information. Models published to the Product Catalog are automatically published to the Service Catalog. The Service Catalog

includes information about goods (models) and services. If the model is available from multiple vendors, a model can be listed more than once. Models are included with Asset Management.

Key features of the Product Catalog use case

This use case lets you include hardware and software product information as items in the Product Catalog.

Product Catalog items

The screenshot shows the 'Product Catalog item' creation screen. At the top, there's a header with a back arrow, a list icon, and tabs for 'General*', 'Product Information', and 'Images'. Below the header are several input fields: 'Name' (empty), 'Application' (set to 'Global'), 'Vendor' (empty), 'Product ID' (empty), 'Model' (marked with a red asterisk, empty), 'Price' (\$0.00), and 'Class' (set to 'Product Catalog Item'). A 'Short description' field is also present. The 'Product Information' tab is active, showing a 'Category' field with a search bar and a note: 'If you want users to be able to search for this item, add it to a Category'. Below this are delivery time settings ('Days 2 Hours 00 00 00') and a large area for images with a grid icon. At the bottom are 'Submit' and 'Try It' buttons.

Related Links

[Deactivate](#)

Results of Product Catalog use case

The CSDM framework ensures that product models are available in the catalog and that there are processes defined to consume the models.

Consider these points while implementing the CSDM framework.

-

The Product Catalog builds on the Service Catalog. Product catalogs specify the product models that are available for use. Product-specific attributes are captured in each catalog item. You can create product catalogs for each type of products in your portfolio.

- You can create catalog types that reflect the types of offered products and services.

CSDM reference

The following sections show the terms, tables, and relationships in the CSDM.

Basics

[CSDM terms](#)

[CI relationships in the Common Service Data Model](#)

[How CSDM concepts map to CMDB tables](#)

Life cycles

[Document life cycle](#)

[Hardware life cycle](#)

[Location life cycle](#)

[Logical life cycle](#)

[Product life cycle](#)

- [CSDM terms](#)

Because most ServiceNow products and Now Platform applications closely align with the Common Service Data Model (CSDM), it's helpful to know common CSDM terms.

- [CI relationships in the Common Service Data Model](#)

For configuration management to be most effective, establish relationships between the objects and configuration items (CIs) in the conceptual CSDM.

- [How CSDM concepts map to CMDB tables](#)

The objects in the conceptual CSDM framework must map to the physical model objects (CIs and CI class tables) in the CMDB.

- [Document life cycle](#)

The CSDM framework provides standard fields and values that you can use to track the life cycle of an asset or a CI. The document life-cycle states represent the overall life cycle of document assets (contracts) and CIs (business process) as related to their products.

- [Hardware life cycle](#)

The CSDM framework provides standard fields and values that you can use to track the life cycle of an asset or a CI. The hardware life-cycle states represent the overall life cycle of hardware assets and CIs as related to their products.

- [Location life cycle](#)

The CSDM framework provides standard fields and values that you can use to track the life cycle of an asset or a CI. The location life-cycle states represent the overall life cycle of a location within common data.

- [Logical life cycle](#)

The CSDM framework provides standard fields and values that you can use to track the life cycle of an asset or a CI. The logical life-cycle states represent the overall life cycle of logical assets and CIs as related to their products.

- [Product life cycle](#)

The CSDM framework provides standard fields and values that you can use to track the life cycle of an asset or a CI. The product life-cycle states represent the overall life cycle of a product model, a specific version, or a product configuration.

CSDM terms

Because most ServiceNow products and Now Platform applications closely align with the Common Service Data Model (CSDM), it's helpful to know common CSDM terms.

Common CSDM terms

Term	Definition	Notes
Application	Any deployed program, module, or group of programs that is designed to provide specific functionality on a computer infrastructure.	Defines behavior and has specific functionality associated with it. Applications are typically discoverable functionality, like Apache Web Server.
Application service	A service type that is a logical representation of a deployed application stack.	Examples of application services are hosting, data backup, and recovery. Note: Applications and application services do not have a one-to-one relationship. An application service can include multiple applications. An application can be included in multiple application services.

Term	Definition	Notes
Asset	An item whose financial value is tracked.	Many assets are CIs and vice versa, but that is not always the case. Assets have a life cycle with financial considerations, for example, Microsoft 365.
Business Application	Represents all software and infrastructure environments (dev, test, prod) that are configured to provide functionality.	Used to increase productivity and perform other business functions accurately. For example, Dell Online.
Business capability	High-level capability that an organization requires to execute its business model or fulfill its mission.	Typically described in the context of performing one or more specific tasks to achieve business outcomes. For example, demand management or financial planning.
Business service	A business service is a service type that is published to business users. A business service typically implements one or more business capabilities.	<p>Usually, business users order business services. Business users can select the desired offering and service commitment levels via the Service Catalog. For example, procurement, shipping, and finance.</p> <ul style="list-style-type: none"> A business service is an operational CI.

Term	Definition	Notes
		<ul style="list-style-type: none"> A business service must be a one-level service and not a hierarchy of business services. A business service can be used for impact in Incident, Problem, and Change and for approvals for Change. A business service must be focused on the consumer or seller.
Configuration item (CI)	Physical and logical components of an infrastructure that are currently or soon will be under configuration management.	Might be a single module such as a server, database, or router or a more complex item, such as a complete system. For example, a web server, database, or infrastructure.
Operating model	An abstract and ideally visual representation (model) of how an organization delivers value to its customers or beneficiaries.	Typically represents the various elements of how an organization operates. It usually incorporates strategy positions such as the innovation model, degree of intelligent automation, industry alignment,

Term	Definition	Notes
		provider delivery models, and the business expectations of IT.
Portfolio	Collection of services, products, projects, or applications.	Used to manage like items together for a business. Portfolios may be grouped by objective, capabilities, organization, like projects, or services.
Service	Means of delivering value to customers by facilitating outcomes that users want to achieve without the ownership of specific costs and risks.	<p>Typically has three aspects:</p> <ul style="list-style-type: none"> • Interaction • Offering • Service system <p>ServiceNow provides three base service types:</p> <ul style="list-style-type: none"> • Business • Technical • Application <p>You can extend the base types to align with the service types in your organization.</p>
Service catalog	Provides consumable view of available products, services,	Helps manage which services a user may have access to. Also, catalogs are

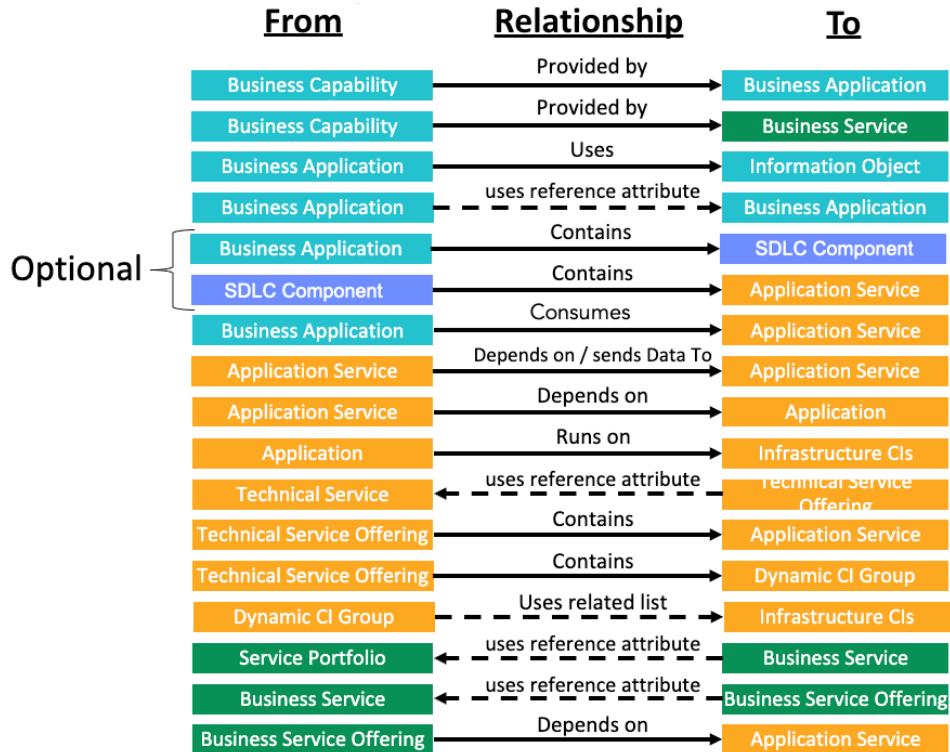
Term	Definition	Notes
	service commitment options, and offerings.	the starting point for access to available services. For example, IT services catalog.
Service commitment	Defines the service delivery obligations agreed to between the consumer and the provider.	Often manifested in the form of contracts such as service level agreements, operational level agreements, and underpinning contracts. Service commitments include specific performance characteristics that differentiate one offering from another.
Service offering	A stratification of a service into capability, availability, pricing, and packaging options.	Different levels of performance and features for a given service can be made available. For example, ITSM Standard and ITSM Pro.
Technical service	A service type that is published to service owners and typically underpins a business or application service.	Typically orderable by service owners. Service owners are able to select the desired offering and service commitment levels via the Service Catalog. For example, computers, storage, and networks).

CI relationships in the Common Service Data Model

For configuration management to be most effective, establish relationships between the objects and configuration items (CIs) in the conceptual CSDM.

Required Common Service Data Model relationships

- Most features and products, such as the Technology Portfolio Management risk assessment and Application Portfolio Management (APM), depend on the relationships.
- The relationships commonly created as part of Service Mapping and Discovery are the standard for infrastructure CIs. If you map the elements manually, be sure to consider how Discovery would have treated them.
- Not all objects in the CSDM conceptual model are CMDB tables, and not all the objects have relationships. You may need to create some of the following required relationships.



Resources in the ServiceNow Community

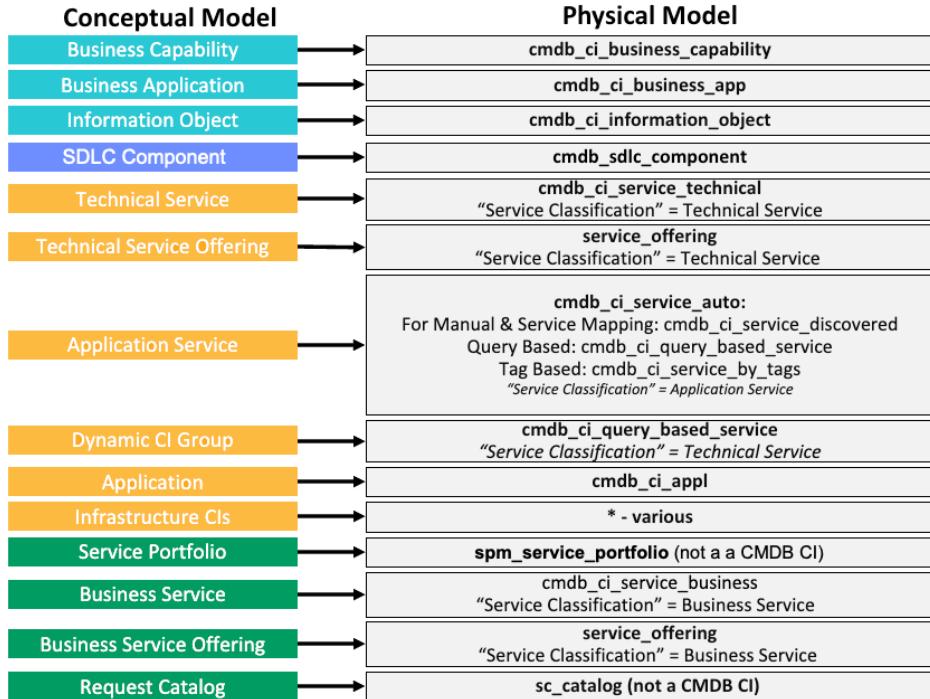
For an extended explanation of the business application reference attribute's role in managing platforms from a CSDM perspective, [see this article](#) in the ServiceNow community.

[CSDM 4.0 What's New](#)

[View the full list of CSDM and data foundation videos.](#)

How CSDM concepts map to CMDB tables

The objects in the conceptual CSDM framework must map to the physical model objects (Clis and CI class tables) in the CMDB.



CSDM videos in the ServiceNow Community

[CSDM 4.0 What's New](#)

[View the full list of CSDM and data foundation videos.](#)

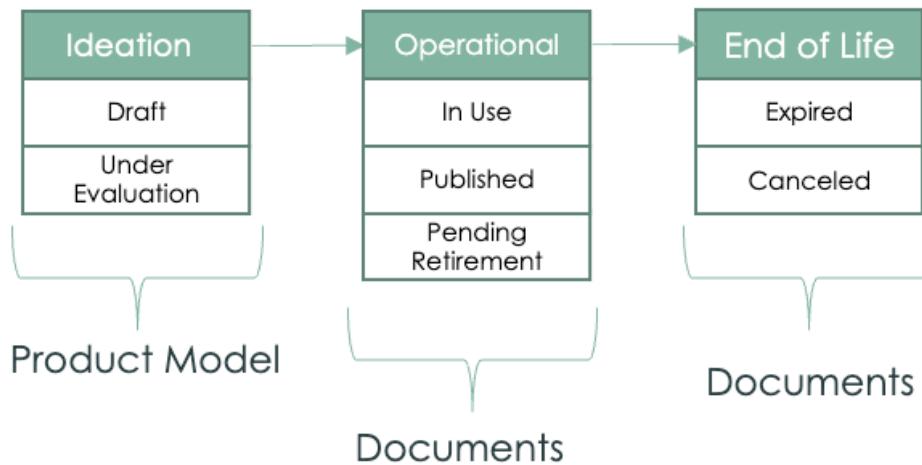
Document life cycle

The CSDM framework provides standard fields and values that you can use to track the life cycle of an asset or a CI. The document life-cycle states represent the overall life cycle of document assets (contracts) and CIs (business process) as related to their products.

Document life-cycle states

A life-cycle state is the combination life-cycle stage and life-cycle status of an asset or CI over the life cycle. For example, a document CI in the **Operational** stage might change status over time from **In Use** to **Published** to **Pending Retirement**. A different document CI in the

Operational stage might go from the **In Use** status directly to **Pending Retirement** status without ever having been in the **Published** status.



The stage and status values for documents are visible only in tables related to documents in Contract Management and the CMDB.

Note: Based on the type of item, the [life_cycle_control] table controls which life-cycle stage values are available for each life-cycle stage.

For additional information on how you can benefit from implementing life cycle states for CMDB entities, see the ['Life cycle states' section in the 'Foundation domain' topic](#).

CSDM videos in the ServiceNow Community

[CSDM V4: Product and life cycle discussion](#)

[CSDM / CMDB discussion: Extending the CMDB and support for the full life cycle](#)

[CSDM 4.0 What's New](#)

[View the full list of CSDM and data foundation videos.](#)

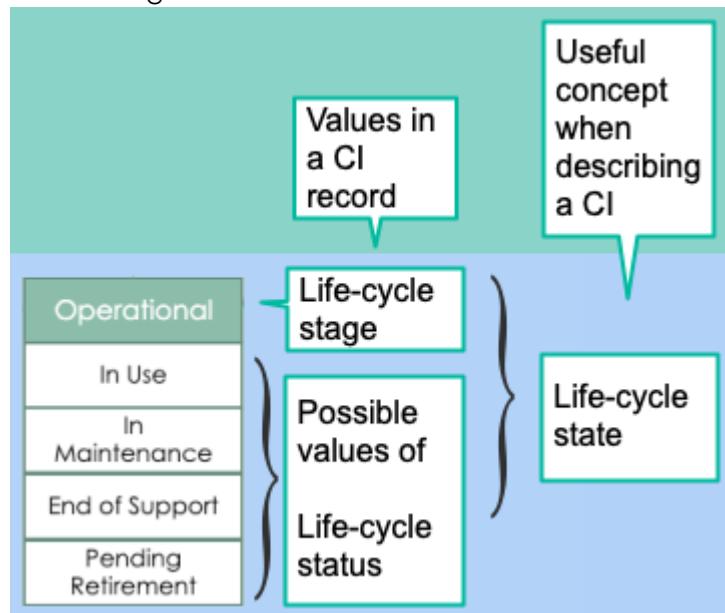
Hardware life cycle

The CSDM framework provides standard fields and values that you can use to track the life cycle of an asset or a CI. The hardware life-cycle states represent the overall life cycle of hardware assets and CIs as related to their products.

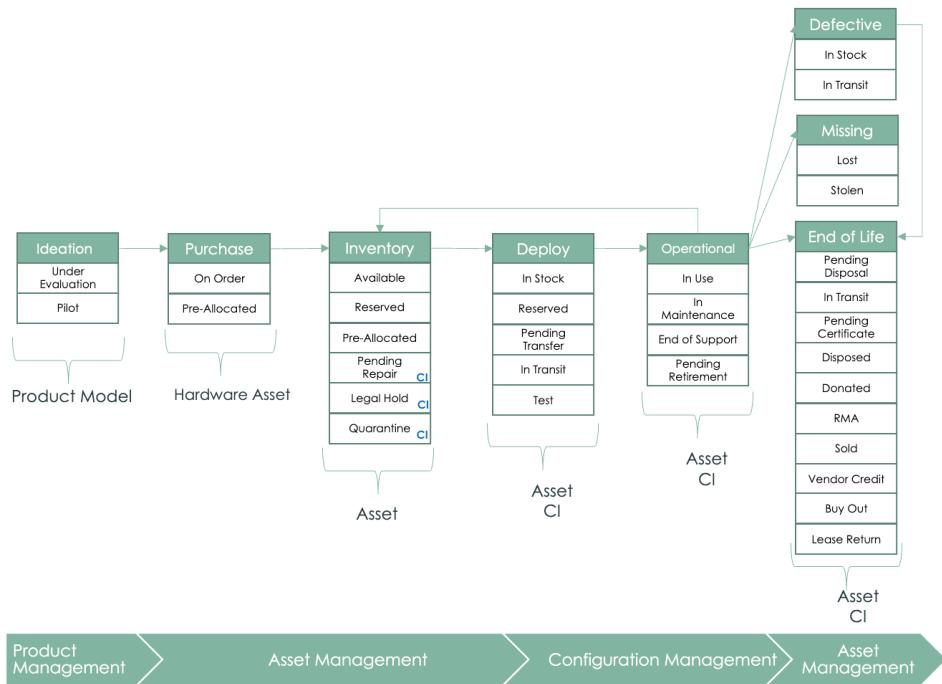
Hardware life-cycle states

Hardware assets are physical items that are stocked, for example servers, monitors, and keyboards. A life-cycle state is the combination life-cycle stage and life-cycle status of an asset or CI over the life cycle. The stages and statuses for the hardware life-cycle process are visible only in hardware-related tables in Asset Management and CMDB.

You can think of the life-cycle state as the combination of two values of an asset or CI: the life-cycle stage and life-cycle status over the CI's life cycle. For example, a hardware CI in the **Operational** stage might change status over time from **In Use** to **In Maintenance** to **End of Support**. A different hardware CI might go from **In Use** to **End of Support** without ever having been in **In Maintenance** status.



Note: Based on the type of item, the [life_cycle_control] table controls which life-cycle stage values are available for each life-cycle stage.



For additional information on how you can benefit from implementing life cycle states for CMDB entities, see the '['Life cycle states' section in the 'Foundation domain' topic](#).

Holistic life cycle: CMDB hardware tables (from cmdb_ci)

CMDB hardware tables	CMDB hardware table name
Accessory	cmdb_ci_acc
Communication Device	cmdb_ci_comm
Computer Peripheral	cmdb_ci_peripheral

CMDB hardware tables	CMDB hardware table name
Computer Room AC	cmdb_ci_crac
Display Hardware	cmdb_ci_display_hardware
Facility Hardware	cmdb_ci_facility_hardware
Hardware	cmdb_ci_hardware
Imaging Hardware	cmdb_ci_imaging_hardware
IP Device	cmdb_ci_ip_device
Monitoring Equipment	cmdb_ci_monitoring_hardware
Network Adaptor	cmdb_ci_network_adaptor
Printing Hardware	cmdb_ci_printing_hardware
Rack	cmdb_ci_rack
Storage Device	cmdb_ci_storage_device

Example hardware classes

View attributes, identification rule, and other important schema structures for the CMDB Computer [cmdb_ci_computer] class. See [Hardware \[cmdb_ci_hardware\] class](#).

How retiring an application service might affect a hardware CI

An application service is the logical representation of the underlying hardware and software CIs that work together to implement a business application or system. The application service represents an instance of the business application or system.

Hardware and software CIs are managed using the physical life cycle states. Because an application service is a logical representation, it is managed as using the logical life-cycle states. The physical hardware CIs that are part of the service map under an application service have their

own life cycle, but they are related through the application services as a specific set of dependencies or decomposition.

Example 1: A hardware CI is not retired when the application service is retired

When an application service is retired, the associated hardware might not be retired. For example, the hardware might remain idle, unrelated to any application service, until it is reallocated for use by a new application service.

Example 2: A hardware CI is shared by multiple application services

In the common scenario of a shared database, multiple application services (each with a unique database schema) share a single database service. The database service runs on a single physical host.

When one of the application services is retired, the database service and host cannot be retired. All of the other application services still depend on the database service that is running on the host.

CSDM videos in the ServiceNow Community

[CSDM V4: Product and life cycle discussion](#)

[CSDM / CMDB discussion: Extending the CMDB and support for the full life cycle](#)

[CSDM 4.0 What's New](#)

[View the full list of CSDM and data foundation videos.](#)

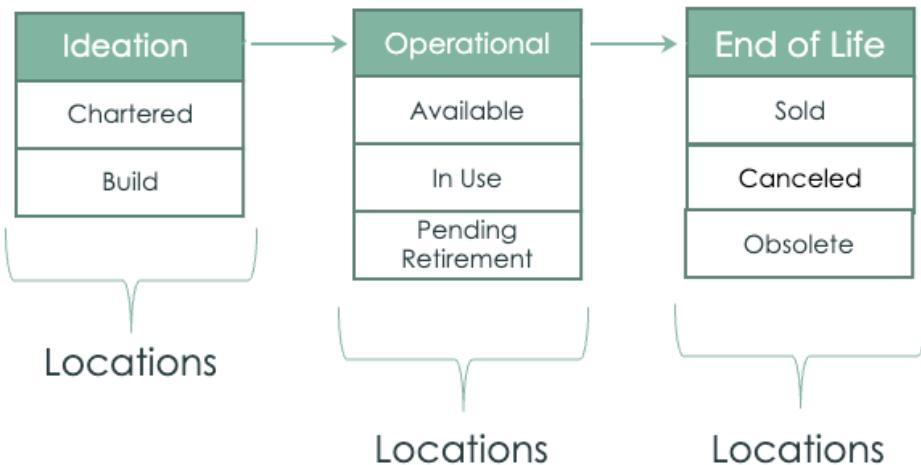
Location life cycle

The CSDM framework provides standard fields and values that you can use to track the life cycle of an asset or a CI. The location life-cycle states represent the overall life cycle of a location within common data.

Location life-cycle states

A life-cycle state is the combination life-cycle stage and life-cycle status of an asset or CI over the life cycle. For example, a location CI in the **Operational** stage might change status over time from **In Use** to **Pending Retirement**. A different location CI in the **Operational** stage might go from

the **Available** status directly to **Pending Retirement** status without ever having been in the **In Use** status.



The stage and status values for locations are visible only in the common data locations table.

Note: Based on the type of item, the [life_cycle_control] table controls which life-cycle stage values are available for each life-cycle stage.

For additional information on how you can benefit from implementing life cycle states for CMDB entities, see the ['Life cycle states' section in the 'Foundation domain' topic](#).

CSDM videos in the ServiceNow Community

[CSDM V4: Product and life cycle discussion](#)

[CSDM / CMDB discussion: Extending the CMDB and support for the full life cycle](#)

[CSDM 4.0 What's New](#)

[View the full list of CSDM and data foundation videos.](#)

Logical life cycle

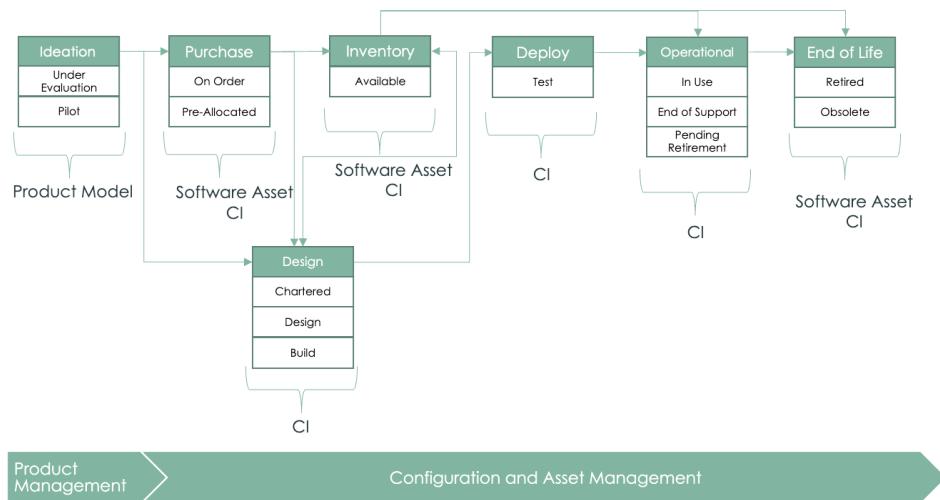
The CSDM framework provides standard fields and values that you can use to track the life cycle of an asset or a CI. The logical life-cycle states represent the overall life cycle of logical assets and CIs as related to their products.

Logical life-cycle states

A logical or software asset includes items like applications, services, and licenses.

A life-cycle state is the combination life-cycle stage and life-cycle status of an asset or CI over the life cycle. For example, a logical CI in the **Operational** stage might change status over time from **In Use** to **Pending Retirement** to **End of Support**. A different logical CI in the **Operational** stage might go from **In Use** status directly to **End of Support** status without ever having been in the **Pending Retirement** status.

Note: A CI may be in the **Operational** stage, but might no longer be supported by the vendor or publisher or third-party. That doesn't mean, however, that it can be or should be retired.



The stage and status values logical items are visible only in tables related to logical items in Asset Management and the CMDB.

Note: Based on the type of item, the [life_cycle_control] table controls which life-cycle stage values are available for each life-cycle stage.

For additional information on how you can benefit from implementing life cycle states for CMDB entities, see the ['Life cycle states' section in the 'Foundation domain' topic](#).

Application service life cycles

Because application services are logical in nature, they should use the Logical life cycle states. Application services follow the same life cycle guidance as any other logical CI.

Holistic Life cycle: CMDB Logical Tables (from cmdb_ci)

CMDB Logical table	Table name
Application Cluster	cmdb_ci_application_cluster
Batch Job	cmdb_ci_batch_job
Business Application	cmdb_ci_business_app
Business Capability	cmdb_ci_business_capability
Business Process	cmdb_ci_business_process
CIM Profiles	cmdb_CIM_profile
Cloud Key Pair	cmdb_ci_cloud_keu_pair
Cloud Mgmt Subnet	cmdb_ci_subnet
Cloud Resource Base	cmdb_ci_resource_base
Cluster	cmdb_ci_cluster
Cluster Node	cmdb_ci_cluster_node
Cluster Resource	cmdb_ci_cluster_resource
Cluster Virtual IP	cmdb_ci_cluster_vip
Configuration File	cmdb_ci_config_file
Database	cmdb_ci_database
Database Catalog	cmdb_ci_db_catalog
Datastore Disk	cmdb_ci_vcenter_datastore_disk
Disk Partition	cmdb_ci_disk_partition
DNS Name	cmdb_ci_dns_name
DRS VM Config	cmdb_ci_drs_vm_config
Endpoint	cmdb_ci_endpoint
Environment	cmdb_ci_environment
Fiber Channel Port	cmdb_ci_fc_port
Information Object	cmdb_ci_information_object
IP Address	cmdb_ci_ip_address
IP Network	cmdb_ci_ip_network
IP Phone	cmdb_ci_ip_phone
IP Service Instance	cmdb_ci_ip_service
Load Balancer Interface	cmdb_ci_lb_interface
Load Balancer Pool	cmdb_ci_lb_pool
Load Balancer Pool Member	cmdb_ci_lb_pool_member
Load Balancer Service	cmdb_ci_lb_service
Load Balancer VLAN	cmdb_ci_lb_vlan
Logical Partition	cmdb_ci_lpar
Memory Module	cmdb_ci_memory_module
Network Appliance Hostname	cmdb_ci_net_app_host
Network Hostname	cmdb_ci_network_host
Network Traffic	cmdb_ci_net_traffic

CMDB Logical table (continued)	Table name
Operating System Level Virtualization Container	cmdb_ci_oslv_container
Operating System Level Virtualization Image	cmdb_ci_oslv_image
Operating System Level Virtualization Image TAg	cmdb_ci_oslv_image_tag
Operating System Level Virtualization Local Image	cmdb_ci_oslv_local_image
Package	cmdb_ci_os_packages
Patch	cmdb_ci_patches
Port	cmdb_ci_port
Print Queue	cmdb_ci_print_queue
Qualifier	cmdb_ci_qualifier
SAN Connection	cmdb_ci_san_connection
SAN Endpoint	cmdb_ci_san_endpoint
SAN Fabric	cmdb_ci_san_fabric
SAN Zone	cmdb_ci_san_zone
SAN Zone Alias	cmdb_ci_san_zone_alias
SAN Zone Alias Member	cmdb_ci_san_zone_alias_member
SAN Zone Member	cmdb_ci_san_zone_member
SAN Zone Set	cmdb_ci_san_zone_set
Service	cmdb_ci_service
Storage Area Network	cmdb_ci_san
Storage Controller	cmdb_ci_storage_controller
Storage Export	cmdb_ci_storage_export
Storage File Share	cmdb_ci_storage_fileshare
Storage HBA	cmdb_ci_storage_hba
Storage Pool	cmdb_ci_storage_pool
Storage Pool Member	cmdb_ci_storage_pool_member
Storage Volume	cmdb_ci_storage_volume
Tomcat Connector	cmdb_ci_tomcat_connector
UPS Bypass	cmdb_ci_ups_bypass
UPS Alarm	cmdb_ci_ups_alarm
UPS Input	cmdb_ci_ups_input
UPS Output	cmdb_ci_ups_output
Veritas Disk Group	cmdb_ci_veritas_disk_group
Virtual Machine Object	cmdb_ci_vm_object
Virtual Private Cloud	cmdb_ci_vpc
Virtual Private Network	cmdb_ci_vpn
Websphere Cell	cmdb_ci_websphere_cell

CSDM videos in the ServiceNow Community

[CSDM V4: Product and life cycle discussion](#)

[CSDM / CMDB discussion: Extending the CMDB and support for the full life cycle](#)

[CSDM 4.0 What's New](#)

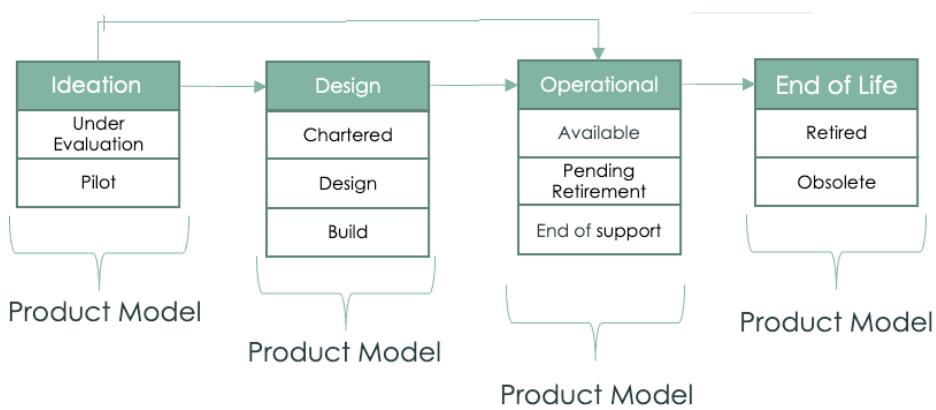
[View the full list of CSDM and data foundation videos.](#)

Product life cycle

The CSDM framework provides standard fields and values that you can use to track the life cycle of an asset or a CI. The product life-cycle states represent the overall life cycle of a product model, a specific version, or a product configuration.

Product life-cycle states

A life-cycle state is the combination life-cycle stage and life-cycle status of an asset or CI over the life cycle. For example, a product CI in the **Operational** stage might change status over time from **Available** to **Pending Retirement** to **End of Support**. A different product CI in the **Operational** stage might go from the **Available** status directly to the **End of Support** status without ever having been in the Pending Retirement status.



Note: Based on the type of item, the [life_cycle_control] table controls which life-cycle stage values are available for each life-cycle stage.

For additional details on products, see [Products and product models](#).

For additional information on how you can benefit from implementing life cycle states for CMDB entities, see the '[Life cycle states](#)' section in the '[Foundation domain](#)' topic.

CSDM videos in the ServiceNow Community

[CSDM V4: Product and life cycle discussion](#)

[CSDM / CMDB discussion: Extending the CMDB and support for the full life cycle](#)

[CSDM 4.0 What's New](#)

[View the full list of CSDM and data foundation videos.](#)

Application services

Understand application services, learn about different application service types and how multiple ServiceNow® business units and products use them.

What application services are

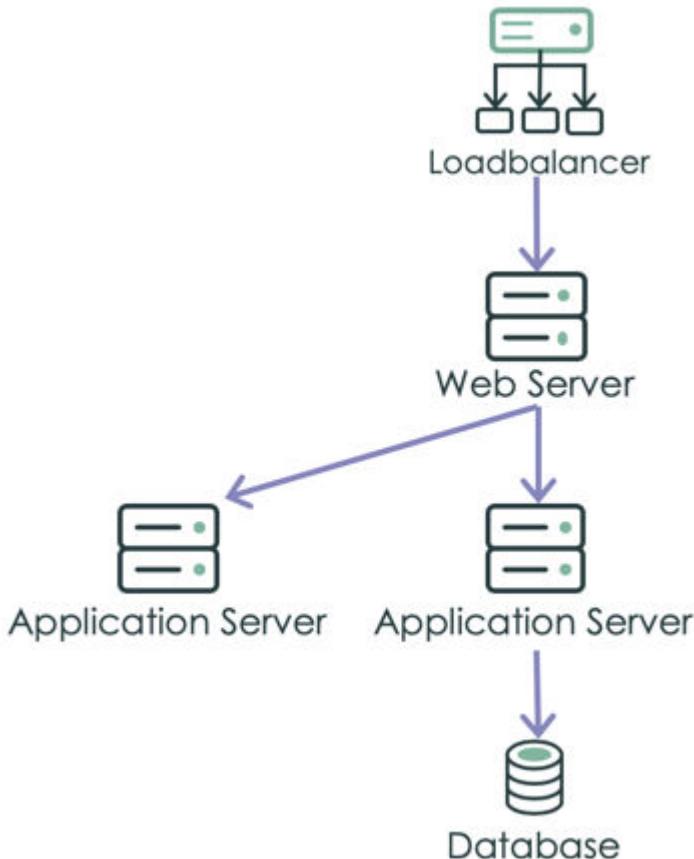
An application service is a set of interconnected applications and hosts which are configured to offer a service to the organization. Application services can be internal, like an organization email system or customer-facing, like an organization website. For example, creating financial reports through a web-based application requires a computer, web server, application server, databases, middleware, and network infrastructure. These applications and hosts are all configured to offer the service of financial reporting. In development environments, application services represent instances of a business application or system in different types of development environments, like development, test, or production.

ServiceNow applications refer to devices and applications that comprise an application service as configuration items (CIs). The various CIs and

the relationships between them, that comprise an application service, are stored in the Configuration Management Database (CMDB).

Each application service contains an entry point as the top-level CI. An entry point is a point where clients access an application service. Typically, it is a URL, or a combination of the IP address and port for application services in enterprise deployments. For cloud-based deployments, an entry point can be a URL to a cloud resource like an AWS gateway.

Application service



The Common Service Data Model (CSDM) helps you streamline service types and service offerings. You can add relationships between application services and other service-related objects in the CSDM:

Business Application, Technical Service Offerings, or Business Service Offerings.

There are the following types of application services:

Discovered

Service Mapping discovers application services using patterns and by following traffic connections.

Pattern-based discovery creates precise and complete application services that represent the service-centric view of the IT infrastructure. It creates a high-fidelity map that is well suited to managing mission-critical application services.

In addition, it provides visibility of cloud-native services such as compute, load balancers, and API gateways. You can use service entry points such as AWS S3 buckets, AWS and Microsoft Azure API gateways, AWS Lambda functions, and Microsoft Azure functions to map services. It can also detect Lambda to Lambda calls and Lambda to RDS connections to build dynamic service maps.

Top-down method maps VMs on-premises and in public clouds. However, it requires these VMs to be fully discovered for the top-down discovery to determine which applications are running in the VM. If a VM isn't fully discovered, use the tag-based method to bridge the gap (see later in this document). Tag-based mapping also maps containers, that you cannot map using the top-down discovery.

Discovered application services have the service classification of application service. They are stored in the Mapped Application Service [cmdb_ci_service_discovered] table.

Dynamic CI Group

Dynamic CI groups which act as application services. The members of the [CMDB groups](#) that is associated with the dynamic CI group, populates the application service. A dynamic CI group is a dynamic grouping of CIs, based on some common criteria such as the location of all web servers in Detroit or all Oracle databases in Boston. After creating a dynamic CI group, it can be used as a group offering in IT Service Management.

If created from the Application Service wizard, the service classification is application service, and if created from the legacy Event Management UI or Service Mapping UI, the classification is technical service.

Application services of the Dynamic CI Group type are stored in the Dynamic CI Group [cmdb_ci_query_based_service] table.

Tag-based

A tag is a label that consists of a key-value pair. Your organization may use tags to categorize its assets, to enhance query and reporting capabilities. Discovery and Cloud Provisioning and Governance can discover tags used by all major cloud providers and container ecosystems. Once the tags are discovered, Service Mapping can create application services based on these tags. For example, you can use tags to map all application services your organization uses in the production environment in the EMEA region.

Tag-based application services have the service classification of application service. They are stored in the Tag-based Application Service [cmdb_ci_service_by_tags] table.

Created Manually

With manual mapping, application owners manually document the applications, IT infrastructure, and relationships that support each application service. This methodology is the best fit for configuration items that are not fully discoverable due to security access issues. For example, IPS devices which support an intrusion prevention service for the security business unit.

Try to avoid manual mapping wherever possible. It's incredibly time consuming to map services manually, and often the information needed for mapping is not available due to evolving technology and a lack of processes that track and document the infrastructure dependencies needed for application context. And, whenever subsequent changes are made to the application service topology, the service map must be manually updated.

Manually created application services have the service classification of application service. Application services of the created manually type are stored in the Mapped Application Service [cmdb_ci_service_discovered] table.

Dynamic

A dynamic application service includes only CIs that are part of CI relationships stored in the CMDB CI Relationship [cmdb_rel_ci] table.

You can't edit a dynamic application service by directly adding or removing CIs from it. Dynamic application services are updated automatically to reflect any change to CI relationships in the CMDB CI Relationship [cmdb_rel_ci] table. When you add a relationship to a CI that is contained in a dynamic application service, then that service automatically updates to reflect the addition of the relationship and the associated new CI. In a similar manner, a dynamic application service automatically updates upon the removal of a relationship and its associated CI from a CI within the service.

One way to create dynamic application services, is by converting legacy business services or legacy manual services (created with Event Management, for example) into application services of the dynamic type.

Dynamic application services have the service classification of application service. Dynamic application services are stored in Calculated Application Services [cmdb_ci_service_calculated] table.

Who uses application services

Application services provide foundation for operation of the following business units and products of the Now Platform:

- **ITOM Health** gathers alerts from infrastructure events captured by third-party monitoring tools. It then uses IT-related information gathered by Discovery to map alerts to configuration items. Based on the collected information, then provides dashboards showing a consolidated view of all service-impact events.
- **ITOM Optimization** gives you tools to provision private and public cloud infrastructure and services and to achieve consistent management and cost visibility. The **Cloud Cost Management** application, available in the ServiceNow Store, helps you to analyze the full range of costs associated with cloud assets so you can identify and take action on opportunities to save money and optimize operations.
- **IT Service Management** users rely on the application services reflecting the IT infrastructure to manage and deliver services to their customers.

- **Customer Service Management** users efficiently diagnose and resolve issues related to the IT infrastructure in the context of application services.
- **Software Asset Management** users understand the software running in your IT environment and track configurations that impact software license consumption across your IT environments and datacenters.
- **Strategic Portfolio Management** users utilize data collected for application services to gain a comprehensive understanding of the applications used in your organization.
- **Security Operations** users view security incidents to find out which application services are at risk. They also use this information to prioritize and resolve threats based on the impact they pose to their organization.

How to create application services

Depending on the needs of your organization, you can deploy different methods of creating and populating application services.

Important: You can use the top-down and manual methods for the same application service. You cannot combine any other methods for creating or populating the same application service.

Analyze the IT infrastructure and service deployment in your organization to pick the optimal method of creating and populating application services.

Choosing the right method for your deployment

Method	When to use	Additional considerations
Top-down discovery Service Mapping performs top-down discovery of application services. Service Mapping uses patterns to discover and map Cls. A pattern is a sequence	Use this method to discover industry-recognized or customized second-tier and third-tier applications. Such applications may include load-balancing solutions,	Pattern-based mapping requires configuring credentials, users, and user permissions to let Service Mapping access applications inside your organization private network. This

Method	When to use	Additional considerations
<p>of steps whose purpose is to detect attributes of a CI and its outbound connections. This method creates precise and complete application services that reliably represent the service-aware view of your organization's IT infrastructure</p> <p>Tag-based discovery in Service Mapping is a complimentary method that enriches the results of top-down discovery.</p>	application or web servers with database connections.	process may take time and effort.
<p>Tag-based</p> <p>If your organization uses tags for asset management, you can use these tags to map application services. Discovery and Cloud Provisioning and Governance discover tags assigned to CIs, and populate the CMDB with this data. Service Mapping uses the tag-related data from the CMDB to map services.</p>	Map resources on cloud workloads like IaaS/PaaS/FaaS/CaaS as well as on container workloads using Kubernetes, OpenShift, or AWS ECS. Also, map resources in the Site Reliability Engineering (SRE) or Customer Reliability Engineering (CRE) deployments. Using tag-based method, you can map container resources in your deployments.	Unlike other mapping methods, tag-based mapping doesn't require configuring credentials or providing users with elevated rights. Tag-based application services may not include relevant CIs, if these CIs don't have correct tags assigned to them.

Method	When to use	Additional considerations
<p>Tag-based service mapping complements top-down service mapping. It provides visibility of containers and also maps VMs that aren't fully discovered, which top-down service mapping is unable to do. However, while tag-based mapping associates tagged components with specific application services, it doesn't map the connections between these components—This is another reason why tag-based mapping complements rather than replaces top-down service mapping.</p>	<p>Typically, you use this method to discover applications on cloud virtualizations or PaaS deployments.</p>	
<p>Ingesting Application Performance Management (APM) maps from integrated Dynatrace or AppDynamics deployments</p> <p>Create application services using the integration with AppDynamics application model and Dynatrace</p>	<p>Use this integration to create application services based on APM maps from Dynatrace or AppDynamics. You are able to use application services created by this method for monitoring Health.</p>	<p>Analyze discovered resources in the CMDB before ingesting from 3rd party to avoid creating duplicate Cls.</p>

Method	When to use	Additional considerations
monitoring platform available on ServiceNow Store.		
<p>Populate an application service using the Dynamic CI Group method</p> <p>Based on CMDB groups, whose members populate the application service.</p>	<p>Use this method as a simple and fast way to create dynamic CI groups for deployments including Microsoft Active Directory, Microsoft Exchange or other DNS-related services. Dynamic CI Groups are especially useful if only a list of resource is available, without configuration details or credentials.</p> <p>Using a CMDB group lets you use CMDB Health to monitor health, and use a CMDB Query Builder saved query to filter for the CIs included in the application service.</p>	<p>There is no map view for application services created using this method. You can only view CIs belonging to such an application service as a list.</p> <p>Need to ensure that the CMDB group accurately filters for the CIs that should be included in the application service.</p>
<p>Application service API</p> <p>Create an automation for creating application services in bulk. Use this method, if your organization has performed cross-organization mapping and analysis and</p>	<p>Use this method for environments that require tracing of DevOps Continuous Integration/Continuous Deployment (CI/CD) process.</p> <p>You can import third-party service</p>	<p>Be familiar with the exact service structure: sys_id of each CI comprising the service and the hierarchy that the CIs form. This method requires knowledge of the scripting infrastructure that your organization uses.</p>

Method	When to use	Additional considerations
<p>collected some information about services.</p> <p>Application services created using APIs belong to the manual type are stored in the Mapped Application Service [cmdb_ci_service_discovered] table.</p>	<p>maps into manual application services individually or in bulk. For example, see the Digital Guidebook: Importing 3rd-party service maps into ServiceNow Service Mapping.</p>	
<p>Populate an application service using the Manual method</p> <p>Create a manual application service with one CI only: the entry point. To populate a manually created application service, add other CIs manually as described in Manually add CIs to an application service.</p> <p>Alternatively, create and populate manual application services by converting business services created in the CMDB and stored in [cmdb_ci_service].</p>	<p>Use the manual method if you can't use other methods of creating or populating application services.</p> <p>Create application services manually for intrusion prevention.</p>	<p>This method doesn't require any preexisting setup or object configuration.</p> <p>You can include CIs of any class in manually created application services.</p> <p>Manually created application services reflect some changes to CIs, like CI attributes. However, they do not automatically reflect changes to CI relationships.</p>
<p>Populate an application service using the Dynamic Service method</p>	<p>Use this method to transform legacy business services into application services</p>	<p>You can't edit a dynamic application service by adding or removing CIs</p>

Method	When to use	Additional considerations
<p>Application services that automatically update to reflect any change to CI relationships in the CMDB CI Relationship [cmdb_rel_ci] table.</p> <p>To conform with Common Service Data Model, you can also convert legacy services to dynamic application services. Those legacy services are stored in the [cmdb_ci_service] or [cmdb_ci_service_manual] CMDB tables:</p> <ul style="list-style-type: none"> • Convert business services to application services • Convert legacy manual services into dynamic application services 	<p>that other ServiceNow products can utilize. For example, dynamic application services can be used for service monitoring and change management.</p>	from it. The system automatically modifies an application service of the dynamic type when you modify relevant relationships for CIs that are part of that application service.
<p>From CSV file</p> <p>Service Mapping extracts information from this file and creates potential application services referred to as service candidates. Use this method, if your organization has performed cross-</p>	<p>If necessary, you can import service candidates from multiple CSV files.</p>	Organize all the collected information in a specific order in a CSV file, precisely as described in the documentation.

Method	When to use	Additional considerations
organization mapping and analysis and collected some information about services.		

To comply with CSDM, convert manual services created using IT Operations Management Event Management and stored in [cmdb_ci_service_manual] as covered in [Convert manual services to application services](#) or [Convert manual services to application services using API](#). Converted services become application services of the manual type stored in the Mapped Application Service [cmdb_ci_service_discovered] table.

Domain separation

Domain separation, if deployed, impacts application services as follows:

- When creating an application service, the application service is assigned to the user's domain.
- When manually adding a CI to an application service, you can choose only CIs that belong to the service domain.
- When using the [createOrUpdateService - POST](#) REST API for creating or updating an application service, the process stops if one of the CIs referenced in the API belongs to a different domain than the application service itself.
- When converting business services into application services, the newly created application service belongs to the same domain as the original business service. The application service comprises only CIs belonging to the same domain as the application service itself.

Create an application service

Create an application service to adhere to CSDM standards and to standardize the organization, maintenance, and monitoring of services in your organization.

Before you begin

Role required: Depending on the population method used:

- Dynamic CI Group: app_service_admin
- Manual: app_service_admin
- Dynamic Service: app_service_admin
- Top Down Discovery: sm_admin
- Tags: sm_admin

About this task

An application service is a set of interconnected applications and hosts which are configured to offer a service to the organization. Application services can be internal, like an organization email system or customer-facing, like an organization website.

An application service has an entry point, which lets users access the application service. If you are at the planning stage and do not know what the entry points are for an application service, you can create the application service without entry points. Such an application service is referred to as an empty application service, to which you can add entry points at any later time.

All application services created in the Application Service wizard, are set with the application service service classification.

Service Mapping, if activated, can automatically discover and map application services as described in [Application service mapping](#). A discovered application service contains the CIs and the connections between them that Service Mapping discovered and mapped.

You can also create an application service by using the [createOrUpdateService - POST](#) REST API.

Procedure

1. Navigate to **All > CSDM > Manage Technical Services > Application Service**.

2. In the Application Services list view, click **New** to open the Application Service wizard.
3. In the **Provide Basic Details** tab:

- a. Fill out the fields for Basic Details.

The screenshot shows the ServiceNow interface for creating a new application service. At the top, there's a navigation bar with links for All, Favorites, History, Workspaces, and Admin. Below that, a breadcrumb trail says 'New Application Service'. A green header bar indicates '1. Provide Basic Details'. The main section is titled 'Basic Details' with a sub-instruction: 'Define the ownership and service level details about this application service.' A table below lists four fields: 'Number' (pre-populated unique ID), 'Name' (unique application service name), 'Environment' (the environment of the offering), and 'Version' (the application service configuration version).

Field	Description
Number	Pre-populated unique ID for the application service.
Name	Unique application service name which is not in use by any other type of application service. Use self-explanatory names such as mailing service or printing service.
Environment	The environment of the offering such as production, development, or test, as identified by some service offerings. Used by Incident Management and Change Management.
Version	The application service configuration version.

Field	Description
Model ID	A product model such as a software model where end of life data is stored.
Operational Status	Operational status of the application service, such as Ready or Retired .
Support Group	Used by Incident Management as the group managing the contract covering the asset.
Change Group	Used by ITSM for routing of change and change-related tasks.
Managed By Group	Group responsible for managing the data.
Owned By	<p>User who is familiar with the infrastructure and applications making up the service. This user is the application service Subject Matter Expert (SME) who provides information necessary for a successful creation of an application service.</p> <p>If the owner name is not listed, create a user with the sm_app_owner role, as the owner. Alternatively, you can choose a user with the sm_admin role.</p>

Note: See [Teams related list](#) for details about the automatic synchronization between the assignment group fields and the Teams related list.

- b. In the Set Relationships section, add relationships between the application service and other components in the CSDM domain.
- c. Click the **Business Application**, **Technical Service Offering**, **Business Service Offering**, or the **Parent Application Service** tab, and add items to the respective **Selected** list.
 - The technical service offering list includes records in the Service Offering [service_offering] table, in which the Service classification attribute is **Technical Service**.
 - The business service offering list includes records in the Service Offering [service_offering] table, in which the Service classification attribute is **Business Service**.
 - The parent application service list includes application service records from the Application Service [cmdb_ci_service_auto] table. Adding a parent application service relationship creates hierarchies and dependencies of application services in deployments such as:
 - Platform host and platform application deployments
 - Micro service deployments in which one or more micro services identified as an application service, is part of a larger application service deployment
 - Shared technical service dependencies
- d. Click **Next**.

For information about CSDM relationships, see [CI relationships in the Common Service Data Model](#).

Also, some fields and relationships are noted as required on the page. To change which fields and which relationships are required, see [Modify the attributes and relationships required for application services](#).

4. On the **Populate the Application Service** tab:



- a. Click **Choose a Method** or click **Next** to skip selecting a service population method.
- b. On the Choose a Method page, select a **Service Population Method**, and then follow the respective link to complete the specific population method:
 - **Top Down Discovery:** Populate application services using top-down discovery
 - **Dynamic CI Group:** Populate an application service using the Dynamic CI Group method.
 - **Tags:** Populate application services using tags
 - **Manual:** Populate an application service using the Manual method
 - **Dynamic Service:** Populate an application service using the Dynamic Service method

Note: The **Top Down Discovery** and the **Tags** options are available only if Service Mapping is installed.
- c. To add another method to populate the application service, click **Add Method** on the Service Population Methods page.. Or, click **Next**.
 - You can add any combination of the **Top Down Discovery** and the **Manual** methods. However, if you select the **Dynamic CI Group**, **Tags**, or the **Dynamic Service** method, the **Add Method** button is grayed out and you cannot add additional methods.
 - You can click a card for a Converted Business Service method to see details about the service conversion, such as the conversion type. For more information, see [Convert business services to application services](#).

5. On the **Preview the Service** tab, review and verify the summaries for creating and populating the application service.



- a. Review **Relationships**.
- b. You can click **Edit Relationships** to modify the relationships to other CSDM objects.
- c. Review **Population Methods Summary**.
- d. You can click **Edit Methods** to modify the selection methods.
- e. Click **Done**.

Result

The application service is created, and you can access the new application service by navigating to application services list views:

- **CSDM > Manage Technical Services > Application Service:** Contains application service Cls from any class extending the Application Service [cmdb_ci_service_auto] class, except alert groups. The list view includes the tag-based, discovered, manual, dynamic CI groups, converted, dynamic, and empty application service types.
- **Configuration > Application Services > Application Services:** Contains application service Cls from any class extending the Application Service [cmdb_ci_service_auto] class, except alert groups. The list view includes the tag-based, discovered, manual, dynamic CI groups, converted, dynamic, and empty application service types.
- **Service Mapping > Services > Application Services:** Contains application service Cls from the Mapped Application Service [cmdb_ci_service_discovered] class. The list view includes the top-down (discovered) and empty application service types.

What to do next

- If the service population method is **Dynamic CI Group**:
 - Click **View CMDB Group CI's** to list all the Cls in the CMDB group that is associated with the application service.

- Click **View Service CI's** to list all the CIs in the application service. Both lists of CIs are identical, unless the CMDB group contains more than 10,000 CIs. In this case, **View CMDB Group CI's** shows all the CIs in the CMDB group, and **View Service CI's** shows only the 10,000 CIs that are members of the application service.
- If the service population method is **Tags**, **Top Down Discovery**, or **Manual**, and click **View Map** to [view the application service map](#) where you can:
 - [Link application services](#)
 - [View CI attributes in an application service map](#)
 - [View the change history of application services](#)
 - [Compare two versions of an application service](#)
- Click **Advanced**, and then on the Advanced Details page, click **Additional Info**, **Questionnaire**, **Reject Messages**, or **Worknotes**, to add details.

Populate application services using top-down discovery

Use top-down discovery to populate an application service. This discovery method deploys discovery patterns to find configuration items (CIs) belonging to the service and connections between these CIs. Pattern-based discovery creates precise and complete application services that reliably represent the service-aware view of your organization's IT infrastructure.

Before you begin

- [Verify that Service Mapping is set up properly.](#)
- Ensure you know which entry point to use for this application service and which attributes you must be able to define for this entry point. Learn about [Entry point attributes](#) available with Service Mapping.
-

Top-Down Discovery is one of several methods for populating an application service with CIs. Choosing a method for populating an

application service, is only one step of the generic procedure for creating an application service. This procedure complements the generic procedure to [Create an application service](#). By itself, this procedure is incomplete.

For information about the different types of application services and the different methods you can use to populate application services, including using top-down discovery, see [Application services](#).

Role required: sm_admin

About this task

A pattern is a sequence of commands whose purpose it is to detect attributes of a CI and its outbound connections. Service Mapping and Discovery share a set of preconfigured patterns that cover most of the commonly used devices and applications.

Service Mapping starts pattern-based top-down discovery process from the entry point you define.

An entry point is a point where clients access an application service. Usually, it is either a URL or a combination of the IP address and port. Service Mapping starts the mapping process from this point. For example, to map your electronic mailing application service, define an IP address or host name of the email server as an entry point.

Entry points vary depending on the nature of the application service. Service Mapping comes with a wide range of preconfigured entry point types that cover many commonly used applications.

Procedure

1. From the **Service Population Method** list in the Choose a Method window, select **Top-Down Discovery**.
2. From the **Application Type** list, select the CI class of the application that serves as the entry point for this application service. Entry point parameters depend on the type you select.
3. Define attributes for the selected entry point as described in [Entry point attributes](#).

4. (Optional) Add free-text comment that may provide useful information for handling this application service later.
5. Click **Save**.

Populate an application service using the Dynamic CI Group method

The Dynamic CI Group method for populating an application service, automatically generates a dynamic CI group. The members of the CMDB group that the dynamic CI group is based on, populates the application service. The application service continuously synchronizes with the CMDB group to reflect any changes in membership in the CMDB group.

Before you begin

The Dynamic CI Group is one of several methods for populating an application service with CIs. Choosing a method for populating an application service, is only one step of the generic procedure for creating an application service. This procedure complements the generic procedure to [Create an application service](#). By itself, this procedure is incomplete.

Note:

- The number of CIs in an application service that is populated by the Dynamic CI Group method, is limited to 10,000, even if the associated CMDB group has more than 10,000 CIs.
- A CMDB group can be used to populate only a single application service. For more information about populating and using CMDB groups, see [CMDB groups](#).
- A dynamic CI group contains CIs but can't contain other groups.

For information about the different types of application services and the different methods you can use to populate application services, including Dynamic CI Group, see [Application services](#).

Role required: app_service_admin

Procedure

1. In the Choose a Method page, select **Dynamic CI Group** as the **Service Population Method**.
2. Fill out the fields that appear, which are specific to the Dynamic CI Group service population method.

Field	Description
Service Population Method	The method used for populating the application service with CIs. Set to Dynamic CI Group .
CMDB Table	Notes the Dynamic CI Group [cmdb_ci_query_based_service] table, in which application services created by the Dynamic CI Group method, are stored.
Group Name	The CMDB group whose members become members the application service. Note: CIs from a class that extends the cmdb_ci_service class (Services), are automatically filtered out and are not added to the application service.

3. Click **Save**.

Result

The alert impact on dynamic CI groups is calculated on the following CIs:

- All CIs that are part of the dynamic CI's CMDB group.
- Children of current CIs with a relationship of: Runs on::Runs
- CIs related to either the current CIs or their children, with a relationship of: Virtualized by::Virtualizes

For more information, see [alert impact calculation](#).

What to do next

Complete the generic procedure [Create an application service](#).

Populate application services using tags

Use tags that help categorize and organize configuration items (CIs) in your organization to populate application services. Tag-based mapping doesn't require configuring credentials or providing users with elevated rights. Tag-based population method requires the Service Mapping feature of ITOM Visibility.

Before you begin

1. Analyze the tag usage in your organization and make a list of all tags and their purposes. Use the Key Value [cmdb_key_value] table to see the tags in the CMDB.
2. If necessary, assign tags to CIs that you want to include in application services.
3. Tags is one of several methods for populating an application service with CIs. Choosing a method for populating an application service, is only one step of the generic procedure for creating an application service. This procedure complements the generic procedure to [Create an application service](#). By itself, this procedure is incomplete.

Role required: sm_admin

About this task

A tag is a label that consists of a key-value pair. Your organization may use tags to categorize its assets, to enhance query and reporting capabilities. Discovery and Cloud Provisioning and Governance can discover tags used by all major cloud providers and container ecosystems. Once the tags are discovered, Service Mapping can create application services based on these tags. For example, you can use tags to map all application services your organization uses in the production environment in the Europe, the Middle East and Africa (EMEA) region.

If you have [configured tag-based service families and tag categories](#), you can use these tag definitions for populating an application service. Part of defining a tag-based service family is defining a tag category, which contains tag keys. If necessary, you can also define tag values to narrow the criteria used for populating application services. Based on the tag definitions for the tag-based service family, Service Mapping creates service candidates - suggested application services. When you use the tag-based service families to populate an application service, you must select the relevant service candidate.

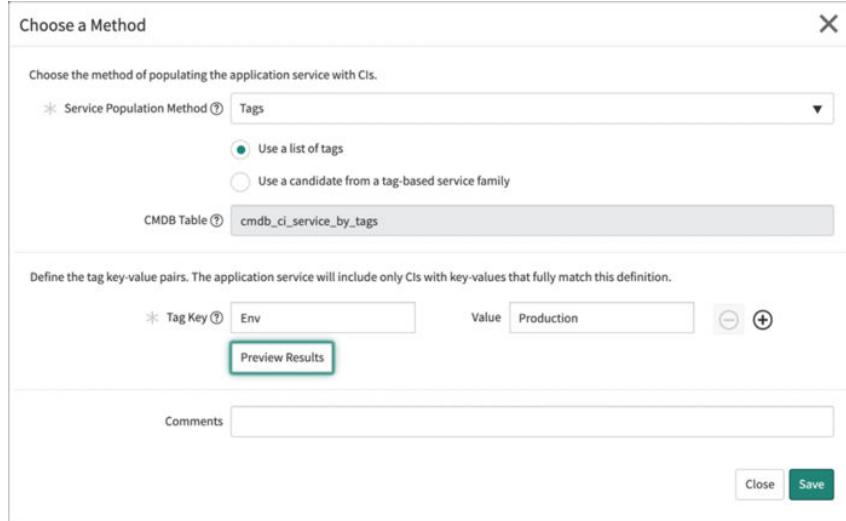
Alternatively, you can define tag keys and their values while choosing the tag-based population method for a new application service. Define up to three tag keys and tag values for the population criteria. CIs that have discovered tag keys and tag values, become part of an application service.

For information about the different types of application services and the different methods you can use to populate application services, including using tags, see [Application services](#).

Note: Service Mapping includes CIs that are part of CI relationships even if these CIs do not have tags assigned to them. For more information, see [Tag-based discovery in Service Mapping](#).

Procedure

1. From the **Service Population Method** list in the Choose a Method window, select **Tags**.
 2. To define new tag criteria, perform the following steps:
 - a. Select **Use a list of tags**.
 - b. Enter the tag key and its respective tag value.
Matching tag keys that exist in the system, appear in the auto-fill options.
- Important:** Tag-based mapping is not case-sensitive; same key names and key values spelled with upper and lower case are identified as the same.
- c. (Optional) Click the plus icon and add another tag key and tag value.



- d. Click **Preview Result** to see the list of CIs that match the defined criteria and if necessary, refine the tag definitions.

Note: You can add no more than three tag key-value pairs for one application service.

3. To use tag definitions from a preconfigured tag-based service family, perform the following steps:
 - a. Select **Use a candidate from a tag-based service family**.
 - b. (Optional) To see the tag definitions for this tag-based service family, click the **Preview** button
 - c. From the **Tag-Based Service Family** list, select the relevant family.
 - d. Review the tag categories and values assigned to the service family.

The dialog box is titled "Choose a Method". It contains the following fields:

- Service Population Method:** Set to "Tags".
 - Use a list of tags
 - Use a candidate from a tag-based service family
- CMDB Table:** cmdb_ci_service_by_tags
- Tag-Based Service Family:** Production-EMEA
- Category:** Environment
- Values:** Production
- Service Candidate:** -- Select one --
- Comments:** (empty)

At the bottom right are "Close" and "Save" buttons.

- e. From the **Service Candidate** list, select the relevant candidate.
- f. (Optional) To review the service candidate form, click the **Preview** button
4. (Optional) Add free-text comment that may provide useful information for handling this application service later.
5. Click **Save**.

Populate an application service using the Manual method

The Manual method for populating an application service, is based on selecting an entry point CI, which lets users access the application service. To populate the application service, you then manually add CIs to the new application service.

Before you begin

Role required: app_service_admin

About this task

Manual is one of several methods for populating an application service with CIs. Choosing a method for populating an application service, is only one step of the generic procedure for creating an application service. This procedure complements the generic procedure to [Create an application service](#). By itself, this procedure is incomplete.

For information about the different types of application services and the different methods you can use to populate application services, including Manual, see [Application services](#).

Procedure

1. In the Choose a Method page, select **Manual** as the **Service Population Method**.
2. Fill out the fields that appear, which are specific to the Manual service population method.

Field	Description
Service Population Method	Manual
CMDB Table	Notes the Mapped Application Service [cmdb_ci_service_discovered] table, in which application services, created by the Manual service population method, are stored.
Class	The class from which to choose the entry point CI for the application service.
CI	The CI from the specified Class , to be the entry point for the application service.

Field	Description
	<p>Note: To eliminate the possibility of delayed results when searching for a specific CI, make your search as specific as possible. A search with *<name> might take a long time and return a large data set.</p>

3. Click **Save**.

What to do next

1. Complete the generic procedure [Create an application service](#).
2. [Manually add CIs](#) to populate the application service.

Populate an application service using the Dynamic Service method

The Dynamic Service method for populating an application service generates a dynamic application service. A dynamic application service automatically updates to reflect any changes to CI relationships in the CMDB CI Relationship [cmdb_rel_ci] table.

Before you begin

The Dynamic Service is one of several methods for populating an application service with CIs. Choosing a method for populating an application service, is only one step of the generic procedure for creating an application service. The following procedure complements the generic procedure to [Create an application service](#). By itself, the following procedure is incomplete.

New dynamic application services initially don't contain any CIs (unless they were converted from legacy business services or legacy manual services). Dynamic application services get automatically populated when they're connected to other CIs with CMDB relationships. Entry

points are automatically created when a relationship between a dynamic application service CI and other CIs is created.

For information about the different types of application services and the different methods you can use to populate application services, including Dynamic Service, see [Application services](#).

Role required: app_service_admin

Procedure

1. In the Choose a Method page, select **Dynamic Service** as the **Service Population Method**.
2. Fill out the fields that appear, which are specific to the Dynamic Service service population method.

Field	Description
Service Population Method	The method used for populating the application service with CIs. Set to Dynamic Service .
CMDB Table	Notes the Calculated Application Services [cmdb_ci_service_calculated] table, in which application services created by the Dynamic Service method, are stored.
Levels	The number of levels of related CIs to include in the application service.

3. Click **Save**.

What to do next

Complete the generic procedure [Create an application service](#).

Monitor the health of application services on the Application Service Dashboard

Monitor the health state of application services in the Application Service Dashboard. With that information, you can reduce the number of incomplete application services by populating them and adding missing data. To use application services effectively, ensure that application services are fully configured and are populated.

Before you begin

Role required: itil_admin or app_service_admin

About this task

Edit the application services that are reported as missing important details. For example, if an application service is not configured with a service population method, configure a service population method for the application service.

The data that appears in the Application Service Dashboard is calculated on a 24-hour cycle during night hours.

Instead of using the legacy Application Service Dashboard that CSDM provides, you can use the latest Application Service Dashboard in the [Insights view in CMDB Workspace](#). The Application services tile in the Insights view provides similar insights as the legacy dashboard, but with a better user experience.

Procedure

1. Navigate to **All > CSDM > Application Service Dashboard**.
2. View counts on the **Application Service Overview** tab, which indicate basic configurations of application services that are incomplete.
 - Count of all application services, complete and incomplete (Total Application Services).
 - Count of application services configured with a service population method (Population method defined).

- Count of application services missing a service population method (No population method defined). Data for the tags and top down discovery methods appear only if Service Mapping is installed.
 - Breakdown of application services by their missing essential data such as an Owner, a Change Group, or a Business Service.
 - Application services without relationships to technical service offerings or to business service offerings.
 - Breakdown of application services by service population methods, including business services that were converted to application services.
3. View counts on the **Related Focus Areas** tab, which indicate other potentially incomplete configurations of application services.
- Total number of application servers, and from those, the number of application servers which are not in any application service. Also, the breakdown of these application servers by class.
 - Total number of databases, and from those, the number of databases which are not in any application service. Also, the breakdown of these databases by class.
 - Total number of hardware servers, and from those, the number of hardware servers which are not in any application service. Also, the breakdown of these servers by class.
- Note:** Application servers, hardware servers, and databases, not included in application services, are counted only up to about 100,000, even if the actual count is greater than this limit. This number limit is determined by the value of the `glide.cmdb.csdm.app_service.max_results` property.
4. (Optional) Filter the count results in either the **Application Service Overview** or the **Related Focus Areas** tab by specific application service properties.
You can filter by the Owner, Change Group, or the Support Group properties. A filter does not impact counts of application services in which the respective filter property is missing. For example, filtering by Owner, doesn't change the count of Missing Owner.

5. (Optional) Select a widget in either the **Application Service Overview** or the **Related Focus Areas** tab to drill down to the respective CI list view.

For example, click the No population method defined widget to see the CI list view of application services in which no population method is defined.

6. (Optional) Edit the coloring rules that determine the threshold for displaying counts in red.

This applies only to the widgets for application services that are not fully configured, such as the No population method defined widget.

- a. Click the **Add Widgets** ('+') icon.
- b. Point to the banner of a widget and click the **Edit Content** icon.
- c. In the Edit Report window, click the **Style** tab and then click **Edit coloring rules**.
- d. See [Create coloring rules for multilevel pivot table reports](#) to complete this edit, and for more details.

Role required for this step is report_admin or admin.

What to do next

After identifying any application services that are missing important details, edit the application services to add the missing details.

1. Navigate to **CSDM > Manage Technical Services > Application Services**.
2. In the Application Services list view, select an application service to edit.
3. Add the missing details.

For details about configuring an application service, see [Create application service](#).

Modify the attributes and relationships required for application services

Modify the lists of attributes and relationships that are required when creating application services.

Before you begin

Role required: itil_admin

About this task

In application service settings, the lists of required attributes and required relationships determine which of those items are required when creating application services. By default, the list of required attributes contains the required Name and Number attributes. Also by default, no relationships are required.

You can choose from predefined lists of allowed required attributes and relationships. To change the requirement status of an attribute, remove it from or add it to the list of required attributes. You can also add Business Application, Technical Service Offering, or Business Service Offering to the list of required relationships.

Procedure

1. Navigate to **All > CSDM > Application Service Settings**.
2. Review the list of **Available** items in the Required attributes and Required relationships lists and then add items to the **Selected** list.
3. Click **Save**.

Result

Next time that you create an application service, the required attributes and relationships are visibly noted in the Basic Details section on the Create an Application Service page.

Convert business services to application services

Unify the way you manage services in the organization by converting manually created records in the Service [cmdb_ci_service] table into application services. Conversion lets you streamline the different types of services in your organization, leverage ITOM Visibility capabilities, and align with the Common Service Data Model (CSDM). The conversion is irreversible: You can't transform application services back into business services.

Using application services has benefits such as:

- Viewing service maps and change history of services.
- Easily seeing the service context by providing a flat list of all CIs in the application service.
- Monitoring service health. If Event Management is deployed, you can monitor service performance and identify health issues for application services.
- In Change Management, the list of impacted services on a change request form is more accurate because the list includes only application services.
- Applying Customer Service Management tools to open and manage cases at the service level.

Discovery doesn't run on converted application services, because converted services are manual. However, if after the conversion you add **Discoverable by Service Mapping** entry points to the application service, then Service Mapping starts discovering such this application service.

Choosing between application services of the manually created and dynamic type

You can convert business services into application services of the manually created type or of the dynamic type. You can edit manually created application services by adding or removing CIs at any time. The system does not update manually created services automatically. If there are changes to CIs making up a manually created application service, the service does not automatically reflect it.

Dynamic services are updated automatically to reflect any change to CI relationships stored in the CMDB CI Relationship [cmdb_rel_ci] table. When you add a relationship to a CI that is contained in a dynamic service, then that service automatically updates to reflect the addition of the relationship and the associated new CI. In a similar manner, a dynamic service automatically updates upon the removal of a relationship and its associated CI from a CI within the service.

To learn more about different types of application services, see [application services](#).

Conversion process

During conversion, the following changes and processes occur:

- The service record is moved from the Service [cmdb_ci_service] table into the Mapped Application Service [cmdb_ci_service_discovered] table by changing the record class.
- The application service is set with all the original business service attributes such as name, owner, and operational status.
- The system adds related items from the business service to the converted application service, up to the specified level.
- The system queries the CMDB for the latest CI changes.
- Event Management, if activated, applies CI impact rules to CIs that are associated with alerts and that are part of the application service. Event Management deploys CI impact rules for alert monitoring.
- You can edit a converted application service of the manually created type by navigating to **CSDM > Manage Technical Services > Application Service**. Then select a converted application service. The service population method for a converted application service, is set to **Converted Business Service**. For more information about editing application services, see [Create an application service](#).

Note: You can't edit a dynamic application service by adding or removing CIs from it. The system automatically modifies an application service of the dynamic type when you modify relevant relationships for CIs that are part of that application service.

Non-compliant CIs

A conversion might involve adding CIs of the following CI types, which cannot be added to an application service:

- cmdb_ci_file_system
- cmdb_ci_network_adapter
- cmdb_ci_storage_device
- cmdb_ci_disk_partition

- cmdb_ci_memory_module
- cmdb_ci_ip_address
- cmdb_ci_storage_pool_member
- dscy_net_base
- cmdb_ci_storage_export
- cmdb_ci_endpoint
- cmdb_ci_translation_rule
- cmdb_ci_qualifier
- cmdb_ci_application_cluster
- cmdb_ci_config_file

If the original business service contains related items belonging to these CI types, then the system does not add such CIs or connections coming from them. If necessary, you can prevent CIs of other CI types from being added to application services by modifying the [Manual CI Inclusions / Exclusions \[svc_manual_ci_exclusions_inclusions.list\]](#) table.

Domain separation

In environments with domain separation, only CIs belonging to the same domain as the application service are added to the application service. If there is a domain hierarchy, CIs must belong to the same child domain as the application service.

Convert business services to application services in bulk

Convert a subset of business services to application services, in bulk and automatically rather than one at a time. Individually select the business services for the conversion, and then convert them into application services.

Before you begin

Role required: app_service_admin, ecmdb_admin, or itil_admin

About this task

Use bulk conversion to convert legacy business services to application services. For the bulk conversion, individually select the business services from the Services list view, typically up to 100 business services in a single conversion. You can create multiple bulk conversion records, each with a different set of business services. However, do not include a business service in more than one bulk conversion.

Note: You can't undo this conversion operation.

Procedure

1. Navigate to **All > Configuration > Services**.
2. In the Services list view, select the services that you want to include in the conversion.
3. In the Actions on selected rows drop down list, select **Bulk Convert Application Services**.
4. Fill out the Bulk Convert Services form.

Field	Description
Name	Pre-populated with "Application Service Conversion: <time stamp>".
Select configuration items	List of services included in the conversion. You can unlock the list and select services to remove from the list. Or, use the Select target record search box to search and add services to the list, by service names.

Field	Description
Levels	The number of levels of related CIs to include in the converted application service.
Update service when CMDB updates	Select this check box to convert the business service into an application service of the dynamic type.

5. Select **Start Conversion**.

What to do next

- Check the status of a conversion: On the Bulk Convert Services form, scroll to the Bulk Convert Services Entries section to see the status (such as Ready or Completed) of a conversion.
- Track the progress of a conversion as it runs: In the navigation filter, enter `cmdb_convert_bulk_services.list` and press the Enter key to see the list of conversions, and their progress.
- (Optional) On a change request form, view affected dynamic services. For example, after you add an affected CI that is associated with a dynamic service:
 1. Navigate to **Change > Open**.
 2. Select a new change request to add affected CIs to.
 3. On the Change Request form, scroll to the Related Links section.
 4. Click the **Affected CIs** tab and then click **Add** to add an affected CI to the change request.
 5. Open the form context menu and select **Refresh Impacted Services**.
 6. Click the **Impacted Services/CIs** tab to see any dynamic services that are associated with the affected CI and that are impacted by the change request.

For more information about affected CIs on a change request, see [Associated CIs on a change request](#).

Convert an individual business service to an application service

Manually convert a specific business service to an application service.

Before you begin

Review the original business service to evaluate it.

- Make sure that all CIs and CI relations are relevant for the future application service. If necessary, change the CI relations in the CMDB.
- Make sure that the original business service doesn't contain more than 5000 CI relations. Application services that contain more than 5000 CI relations cause mapping and monitoring performance issues.
- Decide how many levels of CI relations you are going to use during conversion.

Warning: The conversion is irreversible: You can't transform application services back into business services.

Role required: app_service_admin

Procedure

1. Navigate to **All > Configuration > Business Services**.
2. Select the business service that you want to convert to an application service.
3. Click **Convert to Application Service**.
The Converting to Application Service dialog box opens.
4. Select a number from the **Up to** list, as the number of levels of related CIs to include in the conversion.
The maximum number of levels is eight.
5. Select **Update service when CMDB updates** to convert the business service into an application service of the dynamic type.
6. Click **OK**.

Result

The system adds the CIs from the business service to the converted application service.

What to do next

Open the map for the newly converted application services.

Make sure that the application services aren't too large:

- Service Mapping doesn't offer to view CI list instead of a map for an application service.
- There is no discovery message indicating that the application service is too large: The map does not display the entire service, because it is too large. The number of CI connections exceeded the allowed maximum.

If the service is too large, perform the following actions:

- Review the converted application service to identify CI relations irrelevant or redundant for this service. Remove such CI relations in the CMDB.
- Decide how many levels of related CIs you must include into this application service. If necessary, [change the number of levels used in conversion](#) to reduce the service size.

Related concepts

- [Application services](#)

Convert legacy manual services into dynamic application services

Unify the way you manage services in your organization by converting legacy manual services into dynamic application services. Conversion lets you streamline the different types of services in your organization, leverage ITOM capabilities, and align with the Common Service Data Model (CSDM). The conversion is irreversible: You can't transform application services back into legacy manual services.

Before you begin

Role required: app_service_admin

About this task

You can't edit a dynamic application service directly. Instead, the system automatically updates a dynamic application service to reflect any change to CI relationships in the CMDB CI Relationship [cmdb_rel_ci] table.

During conversion, the following changes and processes occur:

- The service record is moved from the Service [cmdb_ci_service] or the Manual Service [cmdb_ci_service_manual] table into the [cmdb_ci_service_calculated] table by changing the record class.
- The dynamic application service is set with all the original attributes of the legacy manual service such as name, owner, and operational status.
- The system adds related items from the legacy manual service to the converted dynamic application service, up to three levels by default.
- All connections created between CIs in the dynamic application service are endpoint CIs with the relationship uses, implement, or application flow.
- Event Management, if activated, applies CI impact rules to CIs that are associated with alerts and that are part of the application service. Event Management deploys CI impact rules for alert monitoring.

A conversion might involve adding non-compliant CIs, which cannot be added to an application service:

- cmdb_ci_file_system
- cmdb_ci_network_adapter
- cmdb_ci_storage_device
- cmdb_ci_disk_partition
- cmdb_ci_memory_module

- cmdb_ci_ip_address
- cmdb_ci_storage_pool_member
- dscy_net_base
- cmdb_ci_storage_export
- cmdb_ci_endpoint
- cmdb_ci_translation_rule
- cmdb_ci_qualifier
- cmdb_ci_application_cluster
- cmdb_ci_config_file

If the original manual service contains related items belonging to these CI types, then the system does not add such CIs or connections coming from them. If necessary, you can prevent CIs of other CI types from being added to application services by modifying the Manual CI Inclusions / Exclusions [svc_manual_ci_exclusions_inclusions.list] table.

Procedure

1. Navigate to **All > Configuration > Application Services > Application Services**.
2. Ensure that the view is set to the default view.
 - a. Click the List controls icon for the list view.
 - b. Click **View** and then select **Default view**.
3. Open the legacy manual service that you want to convert to a dynamic application service.
4. In the Related Links section on the service form, click **Convert to Dynamic Service**.

Manually add CIs to an application service

Add configuration items (CIs) to manually created application services or to services discovered by Service Mapping. You can edit discovered and manually created application services.

Before you begin

- Verify that the CI type for the CI that you are planning to add, exists. If necessary, create the CI type as described in [Create CI types for Service Mapping and Discovery](#).
- Add CIs to the CMDB for the device or application that you want to add, if necessary. See [Populate the CMDB](#) for more information.

Role required: app_service_admin or sm_admin

About this task

Adding a CI to an application service requires creating a relationship between the new CI and a CI in the application service. You can add CIs to an application service that was created manually, by either:

- Adding a method to populate the application service.

Navigate to **CSDM > Manage Technical Services > Application Service**. Select an application service and then use the **Populate the Application Service** tab to choose a method to populate the application service. For more details, see [Create an application service](#).

- Using the application service service map as described in the steps below.

The default relationship type of the added connection in this case is Depends on::Used by. You can modify this default relationship type by changing the value of the [Components installed with application services](#) property.

Important: You cannot fine-tune or edit tag-based and dynamic services from the map.

Information about the CI in application service, to which you are connecting a new CI, is updated in the CMDB. This information includes the type of the relationship between the CIs. If other application services use the same applicative flow, the CMDB recognizes it and adds the CI you added manually to these application services by analogy. For example, you manually added an IBM WebSphere Message Broker to an IBM WebSphere HTTP Listener in the Bank Customer Portal service. The system also adds this IBM WebSphere Message Broker to the same HTTP Listener in the Bank Internal Portal service, because it uses this HTTP Listener. The same logic applies when you remove a CI you added manually: The system removes it from all application services where you either manually added it or the system added it by analogy.

You can manually connect a CI only to actual CIs existing in the CMDB, not to a visualization of other items on the map such as clusters or boundaries. Also, you cannot add CIs of these CI types to an application service:

- cmdb_ci_file_system
- cmdb_ci_network_adapter
- cmdb_ci_storage_device
- cmdb_ci_disk_partition
- cmdb_ci_memory_module
- cmdb_ci_ip_address
- cmdb_ci_storage_pool_member
- dscy_net_base
- cmdb_ci_storage_export
- cmdb_ci_endpoint
- cmdb_ci_translation_rule
- cmdb_ci_qualifier

- cmdb_ci_application_cluster
- cmdb_ci_config_file

If necessary, you can prevent CIs of other CI types from being added to application services by modifying the [Manual CI Inclusions / Exclusions \[svc_manual_ci_exclusions_inclusions.list\]](#) table.

In environments with domain separation, only CIs belonging to the same domain as the application service are added to the application service. If there is a domain hierarchy, CIs must belong to the same child domain as the application service.

If working with an application service discovered by Service Mapping, manually add a CI:

- To indicate that an application service contains a device or application, which Service Mapping cannot discover. For example, add an A/C unit to the Production Floor service.
- To add a temporary placeholder for a CI, which Service Mapping did not discover. In this case you are planning to perform necessary troubleshooting to ensure that Service Mapping discovers this CI in the future. For example, add an IBM WebSphere Message Broker to the Bank Customer Portal service.
- To create an application service that combines entry points and CIs automatically discovered by Service Mapping with entry points and CIs from the CMDB. After you manually add an entry point, you can update the application service with CIs from the CMDB based on the relationships defined there.

For additional information related to Service Mapping, see [Pattern customization](#) and [Enable traffic-based discovery for CI types or specific CIs](#).

Procedure

1. Open the application service map.
 - a. Navigate to **All > CSDM > Manage Technical Services > Application Service**.
 - b. Select the needed application service.

- c. On the application service page, select **View Map**.
2. If needed, click **Edit** to ensure that the map is in Edit mode.
If Service Mapping is deployed, then in Edit mode, the Discovery Messages section appears below the map.
3. To connect a CI to another CI on the map, right-click the CI to which you want to connect the new CI, and then select **Add a CI**.
4. In the Add A CI dialog box, specify the CI to add:

Field	Description
CI Type	Select the CI type (CI class) for the CI you are adding. Every CI belongs to a CI type which contains a set of attributes configured for this kind of CI, for example, cmdb_ci_appl for applications.
CI Name	Select the CI from the list of CIs of the selected CI type. Note: To eliminate the possibility of delayed results when searching for a specific CI, make your search as specific as possible. A search with *<name> might take a long time and return a large data set.

The CI type list includes only allowed CI types. For example, you cannot add an application cluster.

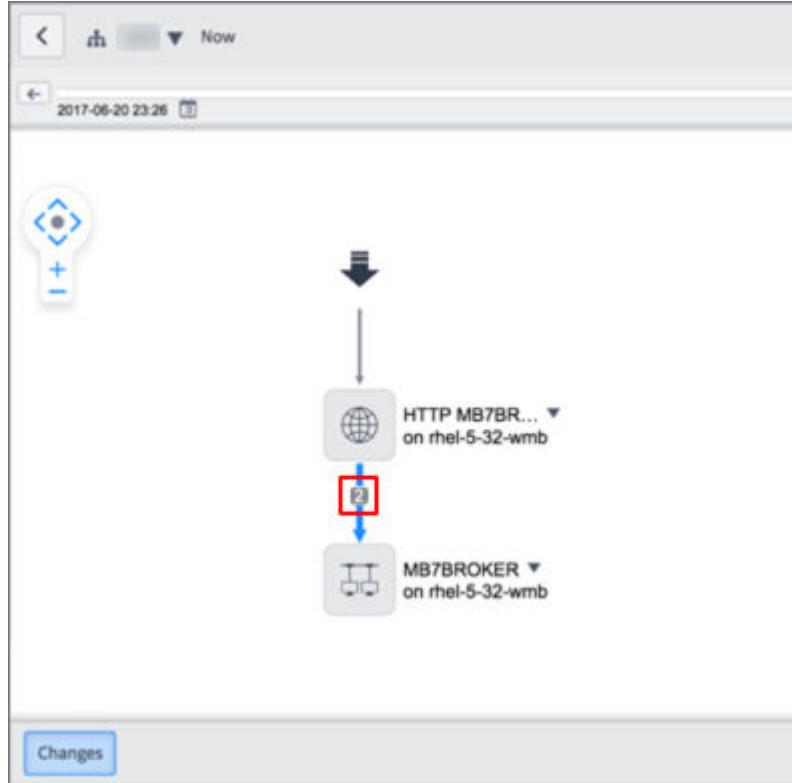
5. Click **Submit**.
The manually added CI appears on the map.

Note: When you manually add a CI, which is an application, as a child to a service that already includes its parent application CI, the newly added child application CI is hidden inside the inclusion. Click the plus (+) symbol next to the parent application CI to see the child application CI.

6. (Optional) If Service Mapping is activated, add a discoverable outgoing connection for the manually added CI:
 - a. Right-click the manually added CI.
 - b. Select **Manually add a connection**.

Note: If you do not see the **Manually add a connection** option in the right-click menu, check that you are logged in with the user that belongs to the same domain as the application service.
 - c. Configure attributes for the entry point as described in [Entry points attributes](#).
 - d. Click **Submit**.

Discovery and Service Mapping attempt to discover this CI. If successful, the CI appears on the map. Otherwise, a warning icon (⚠) appears.
7. (Optional) If Service Mapping is activated and you want Service Mapping to automatically discover a CI, which you previously added manually:
 - a. Customize the relevant pattern or fine-tune traffic-based discovery to enable Service Mapping to discover the CI.
 - b. Navigate to the relevant application service map.
 - c. Click **Run discovery**.
 - d. After the discovery process finishes, verify that Service Mapping discovered the CI by checking the connector leading to the CI. If Service Mapping discovered the CI, then two connectors, a manual and automatically discovered, appear for the CI.



- e. Right-click the CI you added manually.

In the example, it is IBM WebSphere Message Broker.

- f. Select **Remove manually added CI**.

The map shows the CI with only one connector leading to it. If this CI had any manually added connections, they are removed together with the manually added CI.

Related tasks

- [Link application services](#)

Manually update an application service with changes from the CMDB

Ensure that an application service is up-to-date and reflects all the latest changes to its configuration items (CIs). Regularly update application services to reflect any changes to CIs and their relationships in the CMDB.

Before you begin

Role required: app_service_admin

About this task

There is no mechanism or an API that automatically updates application services that were created manually. Also, you may need to manually update application services discovered by Service Mapping, if they contain manually added CIs. You can only update application services which contain manually created entry points and which are not discovered by Service Mapping.

An example of a change is deleting CIs from the CMDB or connecting two CIs one of which is part of an application service. In the first case, your application service may show a CI that no longer exists. In the second case, on the contrary, the application service omits a CI.

An update might involve adding CIs of the following CI types, which cannot be added to an application service:

- cmdb_ci_file_system
- cmdb_ci_network_adapter
- cmdb_ci_storage_device
- cmdb_ci_disk_partition
- cmdb_ci_memory_module
- cmdb_ci_ip_address
- cmdb_ci_storage_pool_member
- dscy_net_base

- cmdb_ci_storage_export
- cmdb_ci_endpoint
- cmdb_ci_translation_rule
- cmdb_ci_qualifier
- cmdb_ci_application_cluster
- cmdb_ci_config_file

If necessary, you can prevent CIs of other CI types from being added to application services by modifying the [Manual CI Inclusions / Exclusions \[svc_manual_ci_exclusions_inclusions.list\]](#) table.

Also, the system can connect a CI from the application service only to actual CIs that exist in the CMDB, not a visualization of other items on the map like clusters or boundaries.

The maximum number of CI connections added to application services during this operation is controlled by the [sa.service_max_ci_service_population](#) property. By default, the value is 1,000 (one thousand connections). Increasing the number of CI connections may cause performance issues. To adjust the maximum number of added CI connections, add the [sa.service_max_ci_service_population](#) property, as described in [Add a system property](#).

In environments with domain separation, only CIs belonging to the same domain as the application service are added into the application service. If there is a domain hierarchy, CIs must belong to the same child domain.

You can also update application services by using [APIs](#).

Procedure

1. Navigate to **All > CSDM > Manage Technical Services > Application Service**.
2. On the Application Services list view, select the application service that you want to update.
3. Click **Advanced** and then click **Advanced Configurations**.

4. On the Additional Info page, click the **Update with changes from CMDB** related link.
5. Select a number in the **Up to** list to limit the number of levels of related items to be updated.
If the specified number is higher than the number of levels of related items that already exist in the application service, then the system adds the missing Cls and their connections.

Warning: Specifying a lower number than the number of levels that already exist in the application service, does not result in the removal of Cls from the application service.
6. Click **OK**.

Result

- The system updates the application service with the changes from the CMDB and shows them on the map.
- After the update process is complete, the application service form opens.

Link application services

You can manually link two application services by adding a reference to one application service into another application service. The service that contains the reference, becomes a dependent service. The service that you include as a reference is a contained service. You can link application services to create dependencies for impact monitoring in Event Management.

Before you begin

You can edit discovered and manually created application services.

Important: You cannot fine-tune or edit tag-based and dynamic services from the map.

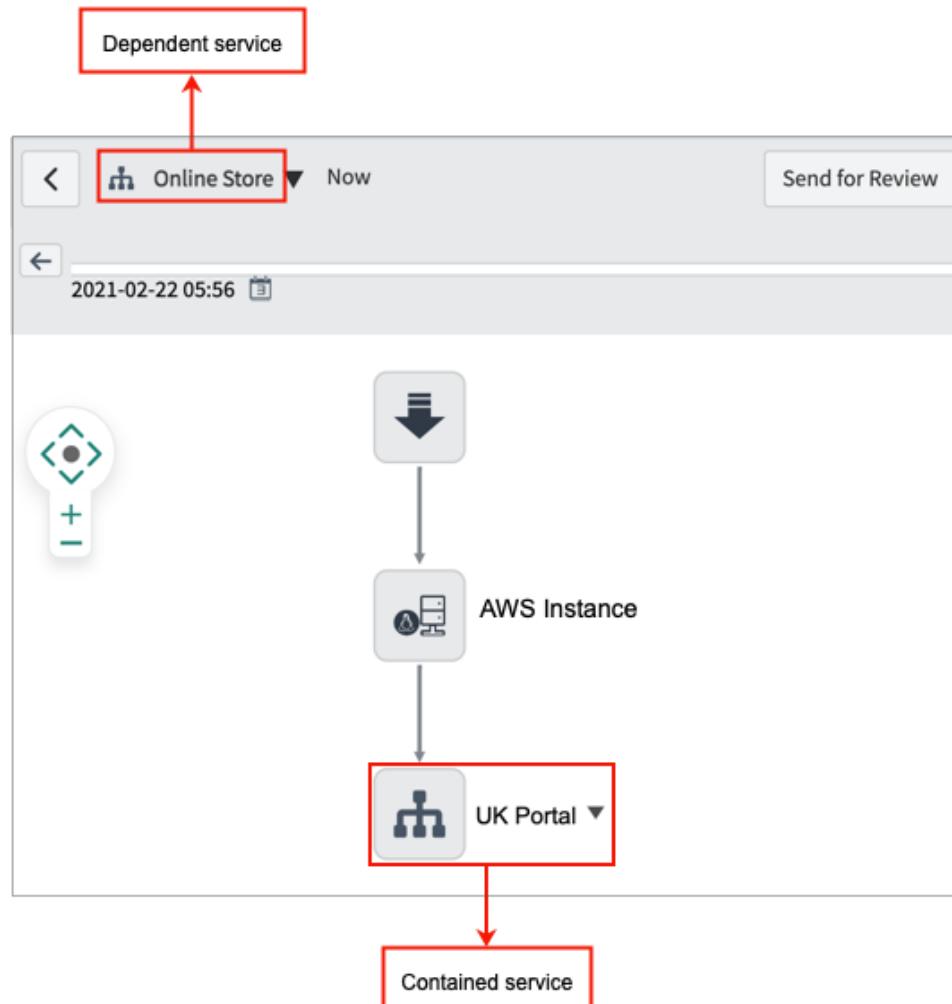
Ensure that you know the name and the service type of the application service, to which you want to add a reference.

Role required: app_service_admin or sm_admin

About this task

To create a link, add a reference to the relevant application service as an outgoing connection of the relevant CI inside another application service. For example, you can add the UK Portal application service as a link to the Online Store application service. In this case, the Online Store service becomes dependent on the UK Portal service that it contains. The Online Store service reflects discovery errors for its contained service in the Edit map mode, as well as alerts in Event Management.

Example of linked application services



When you link an application service to another application service, the information about the CI, to which you linked the service, is updated in the CMDB. The CMDB recognizes other application services that use the same applicative flow, and adds the contained application service to these application services by analogy. The same logic applies when you remove a contained application service: The system removes it from all application services where you either manually linked this service or the system linked this service by analogy.

When using Service Mapping, you may want to link application services to create:

- A dependency between two application services.
- A placeholder for a map branch that Service Mapping failed to discover. If you create or customize a pattern to discover the configuration item (CI) serving as an entry point for the contained application service, Service Mapping can discover this contained service.
- An indication that an application service contains a branch, which Service Mapping cannot discover.

You can add an application service as a contained service to as many application service as necessary.

Procedure

1. Navigate to **Service Mapping > Application Services**.
2. Click **View map** next to the relevant application service.
3. If needed, click **Edit** to ensure that the map is in Edit mode.
4. Right-click the CI to which you want to link an application service as a reference.
5. Select **Add A CI**.
6. In the Add a CI dialog box, select the application service you want to add as a contained service:

Field	Description
CI Type	Select the relevant service type from this list: <ul style="list-style-type: none">• Tag-Based Application Service• Mapped Application Service for discovered or manually created application services• Calculated Application Service for dynamic services• Dynamic CI Group
CI Name	Select the name of the application service that you want to link as a contained service.

7. Click **Submit.**

The icon for the contained service appears on the map.

Related concepts

- [Application services](#)

Group application services

Organize application services by groups to perform actions simultaneously on multiple services, and to control user access to services. You can use Event Management to track service health by service groups.

Before you begin

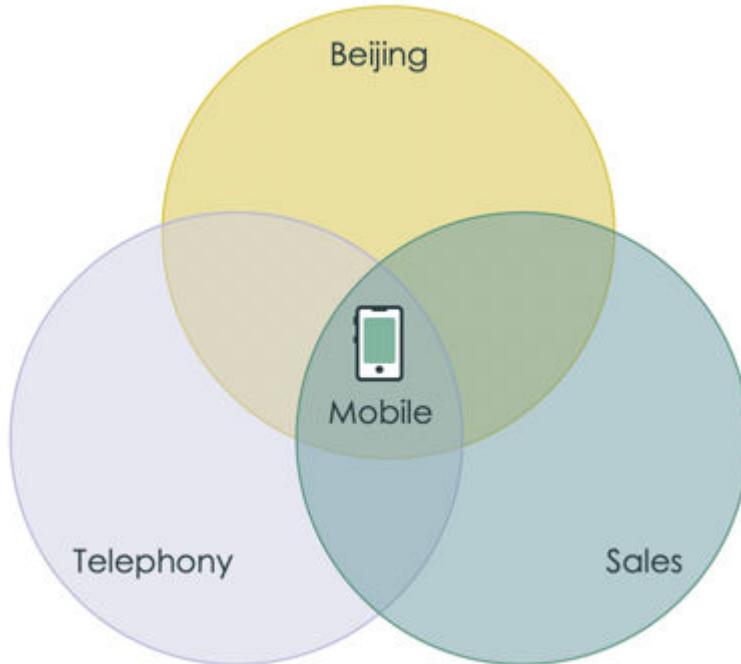
Role required: sm_admin or app_service_admin

About this task

Typically, enterprises have hundreds of services which makes it impractical to manage them individually. Service groups can make service lists much shorter and easier to manage, especially in large organizations or service providers.

How you group application services depends on the user and on service provisioning policies in your enterprise. The relation between application services in groups is purely logical and the same application service can belong to multiple groups. For example, the Mobile service can be part of the following service groups: Sales, Beijing, and Telephony.

Example of an application service belonging to different groups



You can embed a service group within another service group to create a hierarchy of service groups. If users have access to a parent service group, they automatically have access to all its child groups. By default, all new services are assigned to the **All** service group that lets all users view and manage application services. When you assign a role to a service group, the users with this role can access application services in this service group and in the **All** service group. To enable users with this

role to access other services, assign this role to the respective service group. Do not assign user roles directly to the **All** service group.

If Service Mapping is activated, service groups can contain a mixture of manually created application services and application services discovered by Service Mapping.

You can use Now Platform Notifications to alert users if the service group severity changes to critical. The overall severity of the group is determined by the highest alert severity within the group.

Procedure

1. Navigate to **All > Configuration > Application Services > Service Groups**.
2. Click **New**.
3. Enter the name of the new application service group in the **Name** field.
4. To embed this group in another group, enter the name of the other group in the **Parent Group** field.
5. Right-click the form header and click **Save**.
6. Add an application service to the newly created service group.
 - a. In the Service Group Members section, click **New**.
 - b. In the **Name** field, enter the name of the application service. If you are using Event Management, you can also enter an alert group name.
 - c. Click **Submit**.
7. Alternatively, add an application service to a group from the application service form.
 - a. Navigate to **All > Configuration > Application Services > Application Services**.
 - b. Select the application service you want to add to a service group.

- c. In the **Service Group Members** section, double-click **Insert a new row**.
- d. Enter the name of the service group to which you want to add the selected application service.
- e. Click the **OK** icon ().
- f. Click **Update**.

Control user access to application services

Assign user roles to service groups to grant users access to application services in your organization. Your organization may restrict access to some services for security or secrecy reasons.

Before you begin

Make sure that you have performed the user provisioning tasks for the users you want to grant access:

1. [Add users to user groups](#).
2. [Create new roles](#).
3. [Assign roles to users or user groups](#).

Also, make sure that you have created service groups as described in [Group application services](#).

Role required: app_service_admin or sm_admin

About this task

In the base system, the following roles provide access to application services:

app_service_admin

Creates and modifies application services, creates service groups, views, and edits application service maps.

app_service_user

Views maps for operational application services and retrieves service content using the getContent - GET REST API. The itil role that serves as the basic helpdesk technician role contains the app_service_user role.

Service Mapping provides these preconfigured roles:

sm_admin

Sets up the Service Mapping application. Maps, fixes, and maintains application services. Also performs advanced configuration and customization of the product. Assign this role to application administrators.

sm_user

Views maps for operational application services to plan change or migration, as well as analyze the continuity and availability of services. Assign this role to application users.

sm_app_owner

Provides information necessary for successful mapping of an application service. Once a service is mapped, this user reviews the results and either approves it or suggests changes. Assign the sm_app_owner role to users who own application services and are familiar with the infrastructure and applications that make up the services.

Note: Users with the itil role only can view all application services.

Event Management provides these preconfigured roles:

evt_mgmt_admin

Has read and write access to all Event Management features to configure Event Management.

evt_mgmt_operator

In addition to the evt_mgmt_user permissions, can also activate operations on alerts such as acknowledge, close, open incident, and run remediations.

evt_mgmt_user

Has read access to all Event Management features. Has write access to alerts to manage the alert life. Has the itil role to be able to manage incidents that are created from alerts.

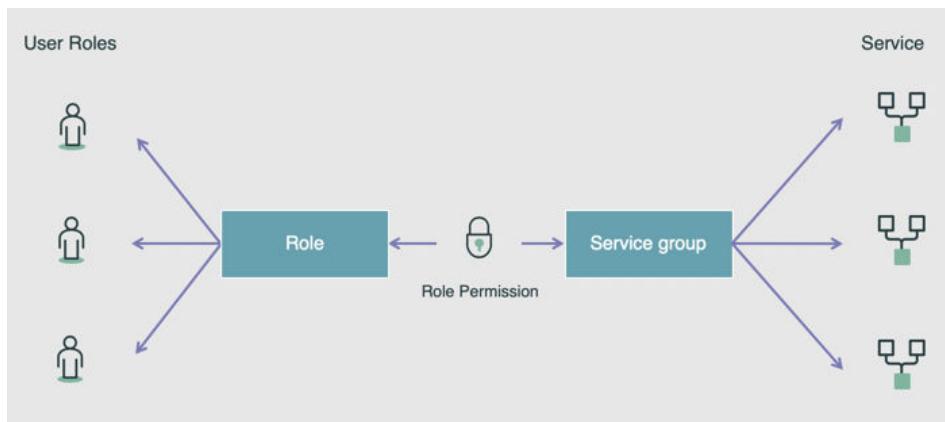
evt_mgmt_integration

Has create access to the Event [em_event] and Registered Nodes [em_registered_nodes] tables to integrate with external event sources.

Typically, enterprises have hundreds of services which makes it impractical to manage them individually. Service groups can make service lists much shorter and easier to manage, especially in large organizations or service providers. In a hierarchy of service groups, access to a parent service group automatically grants access to all the child service groups.

Users inherit permissions from roles that are assigned to them. You can assign some roles directly to service groups to allow all users with this role to access all application services belonging to this group. However, most enterprises choose to organize their roles as a hierarchy. It helps to manage roles across multiple ServiceNow applications. For example, the Service Mapping administrator [sm_admin] can be part of a broader administrator role like administrator [admin]. You can add users to user groups and then assign roles to the user groups to give permissions of this role simultaneously to all the group users.

Assigning a role to an application service group



By default, all new services are assigned to the **All** service group that lets all users view and manage application services. When you assign a role to a service group, the users with this role can access application services in this service group and in the **All** service group. To enable users with this role to access other services, assign this role to the respective service group. Do not assign user roles directly to the **All** service group.

Procedure

1. Navigate to either of the following:
 - **Configuration > Application Services > Service Group Responsibilities.**
 - If Service Mapping is activated: **Service Mapping > Services > Service Group Responsibilities.**
 - If Event Management is activated: **Event Management > Services > Service Group Responsibilities.**
2. Click **New** and fill out the Application Service Group Responsibilities form.

Field	Description
Application Service Group	Service group to which you want to assign a role.
Role	Role you want to assign to the selected service group. For example, financial_services_admin.

3. Click **Submit**.

Example

To manage access to services that contain sensitive financial information in your organization:

1. Organize the services into the Financial Services group.

2. Create a new user role, financial services administrator [financial_services_admin] role, that contains the [app_service_admin] role.
3. Assign the Financial Services administrator role to the Financial Services group.

As a result, only users with the Financial Services administrator role can access application services belonging to the Financial Services group.

View an application service map in base system

An application service map provides a visualization of data for the CIs comprising an application service, and the relationships and connections between these CIs.

Before you begin

Role required: app_service_user to view the map in View mode, and app_service_admin to modify services in Edit mode.

About this task

When you create an application service, the system generates an associated application service map. The system then updates the map to reflect any changes to the application service. This map consists of icons representing CIs and arrows that represent the connections between them.

If Service Mapping is deployed, see [Application service maps](#) and [View CI connection attributes in an application service map](#) for more details.

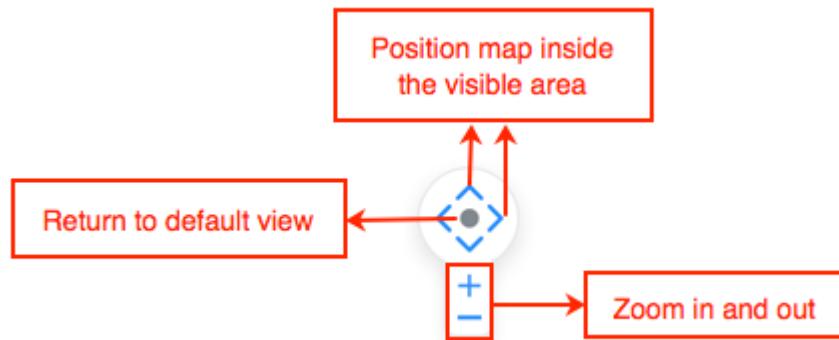
To open an application service map, navigate to **CSDM > Manage Technical Services > Application Service**, select an application service, and then click **View map**.

Perform any of the following operations in the application service map.

Procedure

- Click  on the windows bar to navigate to a different application service.

- Use the navigation tools to increase or decrease the view of the map and to move the map on the page. You can also click anywhere on the map area and drag a segment of the map into the visible area.



- View changes: You can view changes and change records associated with the application service as a whole or with any of its CIs, within a time range. For more information, see [View the change history of application services](#).
Records under the **Change** tab underneath the map, which are associated with a selected CI or connection, are highlighted. If you select a change record under the **Change** tab, then the associated CI icon appears yellow on the map.
- View attributes: When you select a device, application, or connector on the map, it appears in blue and its attributes appear in the Properties pane on the right of the map. When nothing is selected on the map, the details of the application service itself appear on the Properties pane.

Open the CI's form for further details by clicking **Detailed Properties** at the bottom of the Properties pane.

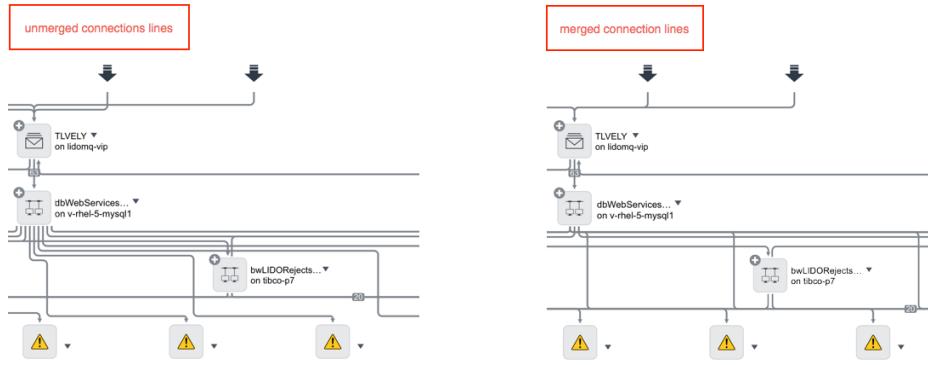
- Click **Edit** or **View** to switch the map mode. Edit mode lets you add or remove CIs from the map.
- Click for **Additional actions**:
 - Set **Group CIs on map**: Simplify maps by grouping 10 or more CIs belonging to the same type and hosted on servers sharing prefix and domain name.

- Set **Spanning tree view**: Simplify the map by organizing CIs into a tree structure and hiding some connection lines. This option is especially useful for very large maps.
- **Map Indicators**: Show additional information for a CI or for the application service itself by displaying related records such as alerts, outages, incidents, and problems. For each indicator that is enabled, the corresponding indicator icon appears next to CIs with associated records, and the corresponding tab appears underneath the map. If a record is associated with the application service itself, the indicator appears next to the application service name.

For information about managing map indicators, see [Create or modify map indicators](#). For more general information, see [Event Management Map Indicators \(Video\)](#).

- **Export to PDF**: Export the map to a .PDF file which you can then share as needed. After the PDF file is ready, click to download the PDF file to your local drive.
- View the details of a connection. By default, connection lines for the same CI on an application service map, are merged. This merge reduces clutter on the map and helps to make the map more readable. For a merged connection line, you can view details for all the underlying connection lines.

Merged connection lines



- To view the source and target CIs of a connection, right-click a connection line.
If spanning tree view is enabled:

- 1: Click the CI whose connections you want to view to show all the concealed connections for the CI.
- 2: Right-click one of the connection lines.
 - To view properties of a connection, click a connection line. For manually added connections, Endpoint Type is Manual Endpoint.
 - To view properties of a connection within a merged connection:
 - 1: Right-click the merged connection line.
 - 2: Select one of the connections.
 - 3: Select **Select edge**.

What to do next

You can change the details that appear in the Properties pane by updating the form view 'Form view and section', as described in [Configuring the form layout](#).

Related reference

- [Spanning tree view property](#)

View CI attributes in an application service map classic Service Mapping

An application service map displays attributes for each configuration item (CI) that is part of the application service, as well as for the application service itself. The attributes come from the CMDB.

Before you begin

Role required: sm_admin or sm_user

About this task

You can view the following information for each CI:

Name label

The CI name. This attribute is either preconfigured on the CI or configured during CI installation.

Basic attributes

A summary of the most important CI attributes.

Detailed attributes

A complete list of all attributes collected for the CI.

Each CI type (CI class) has different attributes. For example, the Linux Server type has different attributes than the SQL Instance type.

If Service Mapping is deployed, the way CIs appear on the map depends on the [view you select for the map](#). Attributes available for viewing also depend on the Service Mapping setup. For more information, see description of [components installed with Service Mapping](#).

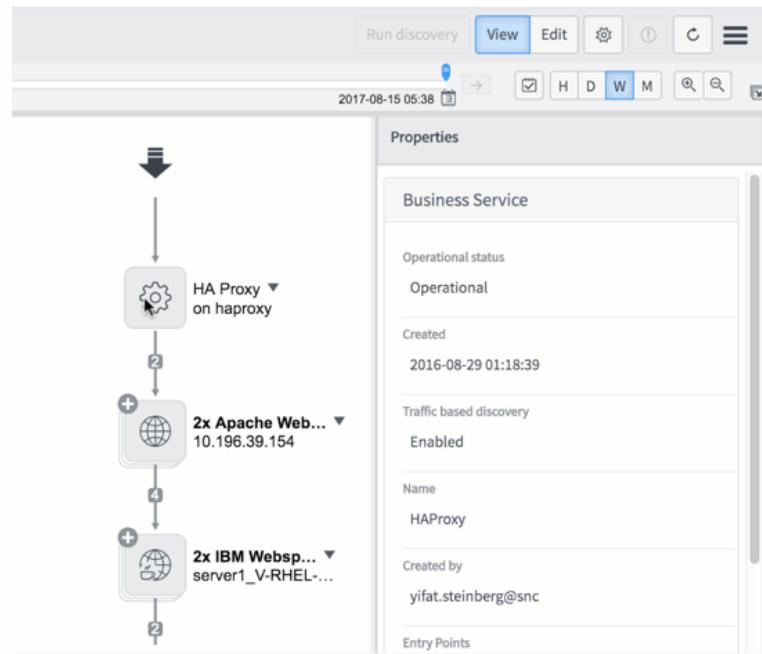
Procedure

1. Open the application service map.
 - a. Navigate to **All > CSDM > Manage Technical Services > Application Service**.
 - b. Select the needed application service.
 - c. On the application service page, select **View Map**.
2. If needed, click **Edit** to ensure that the map is in Edit mode.

If Service Mapping is deployed, then in Edit mode, the Discovery Messages section appears below the map.
3. To see the full name of a CI whose name has been shortened on the map, point to the CI.

A tooltip displays the full CI name.
4. Click a CI to see its details in the **Properties** pane.

The attributes of applications and the servers that host them appear separately.



5. To view more detailed attributes for the CI, click **Detailed properties** at the bottom of the Properties pane.
6. (Optional) To view configuration files associated with a CI in environments where Service Mapping or Discovery are enabled and tracking changes to CI configuration files is enabled:
 - Review the list of files under **Tracked Files** in the **Properties** pane. Click the file name to open the actual file.
 - Click the **Affected CIs** tab and view the list of configuration files. Click the file name to open the actual file.

View the change history of application services in classic Service Mapping

You can view the changes made to an application service as a whole and to the individual configuration items (CIs) comprising the service. Change history is useful for maintenance, planning, or troubleshooting procedures.

Before you begin

Role required: admin, sm_admin, sm_user, app_service_admin, or app_service_user

About this task

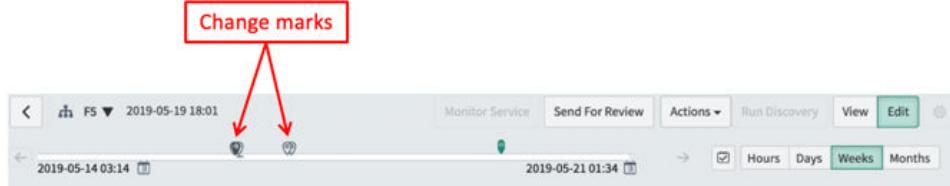
Details about changes to an application service and to its CIs are stored in the CMDB. Typically, these changes reflect adding or removing CIs from an application service, upgrading or updating CIs, or modifying CI configuration files. The system gathers this data by querying CMDB tables and then creating the change history view. In deployments where Service Mapping is activated, the type of change information Service Mapping queries depends on discovery patterns that Service Mapping uses to discover CIs.

Changes to configuration files are associated with CIs to which these files belong. Maps show configuration file changes as changes to related CIs.

While you can see change records for a specific CI in the context of application services, you can also see detailed history of a specific CI separate from its application service as described in [History Timeline](#).

If the Now Platform is configured to validate changes, all changes are evaluated and rendered as valid or not. If a change is valid, its change record on the application service map is marked as approved. For more information about configuring the platform for change validation, see [Managing proposed changes](#).

Changes to the application service appear on the history timeline.



The type of change mark depends on the nature of changes that it represents:

Light gray balloon (📍)

Unapproved change that does not influence the application service behavior. For example, a change in a network path or adding a node to a cluster.

Dark gray balloon (📍)

Unapproved change that changes the application service behavior.

Green balloon (📍)

An approved change in deployments where the Now Platform is configured to validate changes.

Double balloon (📍)

Multiple separate changes that happened a short time from each other.

You can mark times on the history scale by creating baselines to quickly return to the marked view.

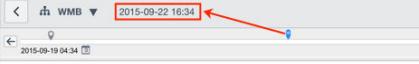
Procedure

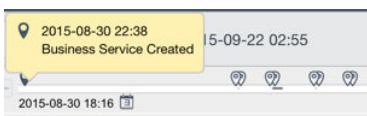
1. Open the application service map.
 - a. Navigate to **All > CSDM > Manage Technical Services > Application Service**.
 - b. Select the needed application service.
 - c. On the application service page, select **View Map**.

2. Review change records created for this application service on the **Changes** tab at the bottom of the page.

If Service Mapping is deployed, then in Edit mode, the Discovery Messages section appears below the map.

3. On the history timeline, set the time range of changes that you want to view.

Option	Action
To set the time range of the history timeline	Click the hour, day, week, or month icons. 
To increase or decrease the time range	Click the zoom in and zoom out icons. 
To change the upper limit on your history range	Click the history scale.  <p>The time that serves as the upper limit appears above the history timeline.</p>

Option	Action
	<p>Note:</p> <p>You cannot set the lower limit on your history range to a time before this application service was created. This time is marked with the IT Service Created event on the history timeline.</p> 

The map shows the history view of the application service for the time you selected.

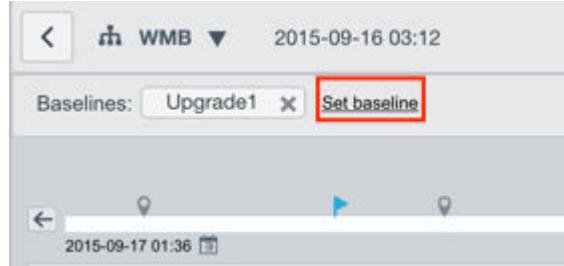
Note: The **Change** tab shows all change records, even the ones which are filtered out of the history view.

4. To mark a time on the time scale, set a baseline:

- Click the **Compare** icon.

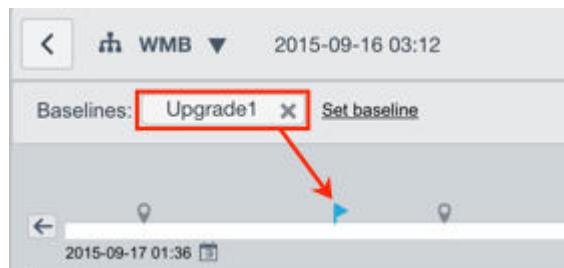


- Navigate to the time you want to mark as a baseline on the history scale.
- Click **Set baseline**.



- d. Enter the name of the baseline and click **OK**.

The new baseline appears as a button above the history scale and as a blue flag on the history scale.



5. View the change history.

Option	Action												
To see the CI responsible for a change record	<p>Select a change record on the Changes tab.</p> <p>The related CI is marked yellow in the map.</p> <p>Changes</p> <table border="1"> <thead> <tr> <th>Created</th> <th>Name</th> <th>Attribute Description</th> </tr> </thead> <tbody> <tr> <td>2015-08-05 03:12:35</td> <td>CI Added</td> <td>CI IIS Virtual Directory was added</td> </tr> <tr> <td>2015-08-05 03:12:35</td> <td>CI Added</td> <td>CI Windows Service was added</td> </tr> <tr> <td>2015-08-05 03:12:35</td> <td>CI Added</td> <td>CI Windows Server was added</td> </tr> </tbody> </table>	Created	Name	Attribute Description	2015-08-05 03:12:35	CI Added	CI IIS Virtual Directory was added	2015-08-05 03:12:35	CI Added	CI Windows Service was added	2015-08-05 03:12:35	CI Added	CI Windows Server was added
Created	Name	Attribute Description											
2015-08-05 03:12:35	CI Added	CI IIS Virtual Directory was added											
2015-08-05 03:12:35	CI Added	CI Windows Service was added											
2015-08-05 03:12:35	CI Added	CI Windows Server was added											

Option	Action						
To see only change records related to a CI	<p>Select the required CI or the connection on the map.</p> <p>The Changes tab displays only change records related to the selected CI or connection.</p> <p>Changes</p> <table border="1"> <thead> <tr> <th>Created</th> <th>Name</th> <th>Attribute Description</th> </tr> </thead> <tbody> <tr> <td>2015-08-09 07:46:26</td> <td>CI Added</td> <td>CI Oracle12A was added</td> </tr> </tbody> </table>	Created	Name	Attribute Description	2015-08-09 07:46:26	CI Added	CI Oracle12A was added
Created	Name	Attribute Description					
2015-08-09 07:46:26	CI Added	CI Oracle12A was added					
To see the configuration file at the selected moment in the past	<ol style="list-style-type: none"> Set the time on the history scale. In the Properties pane, scroll to Tracked Configuration Files, and click the file name. <p>The new tab opens displaying the content of the tracked configuration file at the selected time.</p>						
To see the network at the selected moment in the past	<ol style="list-style-type: none"> Set the time on the history scale. Right-click the connection and select Show network path. 						

Option	Action
	<p>The new tab opens displaying the network or storage path map for the time you selected.</p> <p>Note: You cannot view the network path for connections marked as boundaries to this application service.</p>

6. To exit the history view and see the current status of the application service, click the current icon.



Related tasks

- [View an application service map in base system](#)
- [Compare two versions of an application service in classic Service Mapping](#)

Compare two versions of an application service in classic Service Mapping

You can see a summary of application service changes at a glance by comparing two versions of an application service. This feature is useful for checking the application service status before and after a certain change or problem.

Before you begin

Role required: admin, app_service_admin, app_service_user, sm_admin, or sm_user

About this task

Specify two points in time for which to compare the two versions of an application service. You can use the change indicators on the timeline to specify one point in time that is before and another that is after a change for which to see the details. For example, if you know that the application service started to fail at a certain time, you can compare two versions of the application service, one before and one after the problem started. This comparison lets you see the summary of changes that possibly led to the problems.

Service Mapping, if deployed, tracks and shows all changes to a CI including configuration files associated with a CI. When you compare two versions of an application service, you can see changes made to configuration files as changes to CIs. You can also compare two versions of a configuration file to see the actual changes in the files, during the time range specified for the comparison.

Procedure

1. Open the application service map.
 - a. Navigate to **All > CSDM > Manage Technical Services > Application Service**.
 - b. Select the needed application service.
 - c. On the application service page, select **View Map**.
2. If needed, click **Edit** to ensure that the map is in Edit mode.

If Service Mapping is deployed, then in Edit mode, the Discovery Messages section appears below the map.

3. On the history timeline, set the time range of changes that you want to view.

Option	Action
To set the time range of the history timeline	Click the hour, day, week, or month icons. 

Option	Action
To increase or decrease the time range	Click the zoom in and zoom out icons. 
To change the upper limit on your history range	Click the history scale.  The time that serves as the upper limit appears above the history timeline. <p>Note:</p> <p>You cannot set the lower limit on your history range to a time before this application service was created. This time is marked with the IT Service Created event on the history timeline.</p> 

The map shows the history view of the application service for the time you selected.

Note: The **Change** tab shows all change records, even the ones which are filtered out of the history view.

4. Click the **Compare** icon.

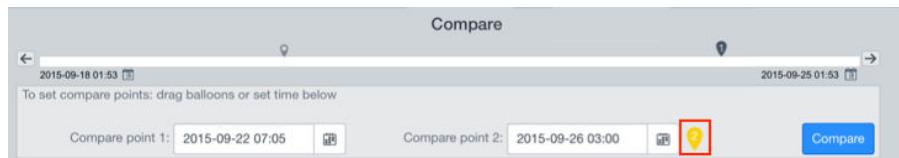


5. Set **Compare point 1** and **Compare point 2** as the two points in time for the comparison.

You can drag the pointers on the history scale to set corresponding time points.



If the history scale does not include the time set for comparison, then its corresponding pointer appears next to the compare point in yellow:

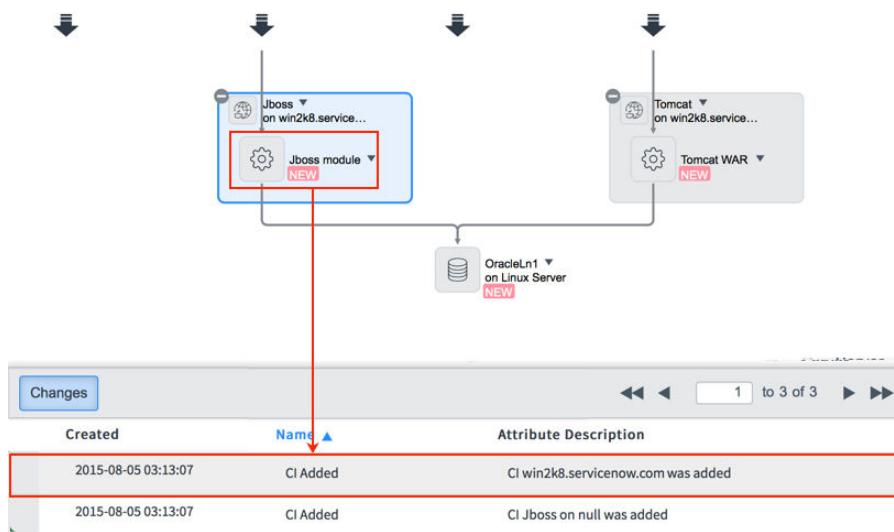


Note: If there are no changes to the service during the time interval specified by **Compare point 1** and **Compare point 2**, then no change details are displayed.

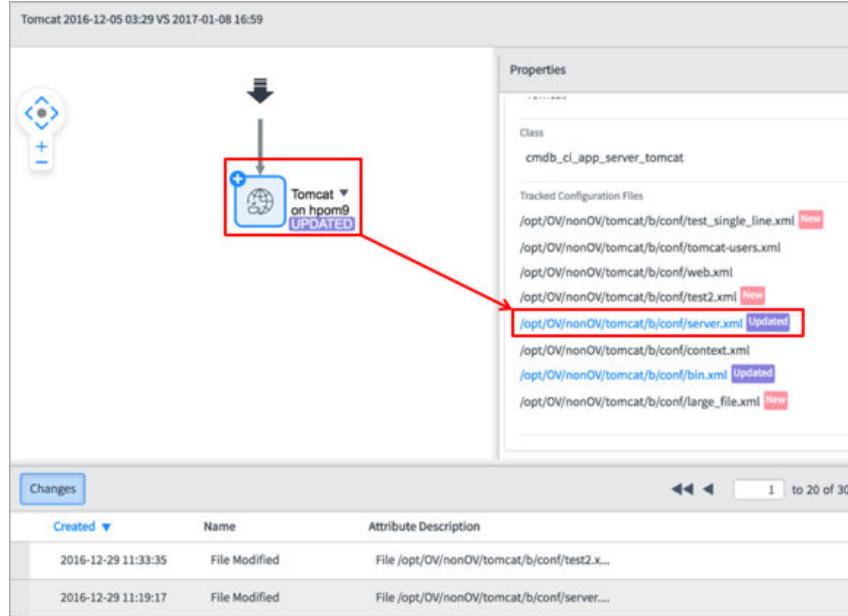
6. Click **Compare**.

The comparison view opens in a separate tab.

7. Select a marked CI to see the relevant change record on the **Changes** tab.



8. (Optional) If Service Mapping is deployed, you can compare two versions of a configuration file that appears on the map as Updated:
 - a. Select the CI that is associated with the updated configuration file.
 - b. In the **Properties** pane, click the link to the updated file.



The **Tracked Configuration Files Version Compare** tab opens showing two versions of the configuration file side by side.

- Review actual changes.

Highlight colors indicate the type of change:

- Purple — Updated line
- Pink — New line
- Gray — Deleted line

- Navigate between the changes using the arrows in the upper right corner.
 - Close the **Tracked Configuration Files Version Compare** tab when finished.
9. Close the comparison view when finished.

Related tasks

- [View the change history of application services in classic Service Mapping](#)

Use application services APIs

Application services provide APIs that let you perform operations such as creating and updating an application service, populating it with CIs from the CMDB, and retrieving details from an existing application service.

Role required: app_service_admin

An application service is a set of interconnected applications and hosts which are configured to offer a service to the organization. Application services can be internal, like an organization email system or customer-facing, like an organization website.

Create an application service

Using the [createOrUpdateService - POST](#) REST API to create an application service suits your organization if the ServiceNow CMDB already contains the CIs making up the service. Typically, it is the case when you have manually added CIs directly into the CMDB, or used the Discovery application to discover CIs and store information about them in the CMDB. You can also use this API to create an application service containing CIs discovered using non-ServiceNow applications.

By default, when an application service is created, all CI connections are of the Depends on::Used by relationship type. You can modify this default type by changing the value of the [sa.it_service.manual_ci_rel_type](#) property.

Before creating an application service, ensure that:

- The CMDB contains all the CIs comprising the application service.
- You have the sys_id of each CI comprised in the application service you want to create.
- You understand the hierarchy that the CIs form.

The Mapped Application Service [cmdb_ci_service_discovered] table contains all application services including services you create using APIs.

You can also manually create an application service using the user interface as described in [Create an application service](#).

Retrieve content from an application service

Use the [getContent - GET](#) REST API to retrieve a list of CIs and the relationships between them, for an application service that was created manually.

Additional APIs

The following JavaScript APIs are also available:

- [`addCI\(\)`](#): Add a CI to a manually created an application service.
For restrictions on the CIs being added and other details about adding CIs to an application service, see [Manually add CIs to an application service](#).
- [`addManualConnection\(\)`](#): Add a manually created connection to an application service.
- [`migrateManualToApplicationService\(\)`](#): Convert a manual service to an application service.
- [`populateApplicationService\(\)`](#): Populate an application service with CIs and relationships from the designated entry point.
- [`removeCI\(\)`](#): Remove a manually created CI from an application service.
- [`removeManualConnection\(\)`](#): Remove a manually created connection and the connected CI from an application service.

Components installed with application services

Several types of components are installed with activation of the Application Service [com.snc.cmdb.it_service] plugin, including tables, user roles, and scheduled jobs.

Note: The Application Files table lists the components that are installed with this application. For instructions on how to access this table, see [Find components installed with an application](#).

Roles installed

Role title [name]	Description	Contains roles
[app_service_user]	Views maps for operational application services and retrieves service content using the getContent - GET REST API. The itil role that serves as the basic helpdesk technician role contains the app_service_user role.	None
[app_service_admin]	Creates and modifies application services, creates service groups, views, and edits application service maps.	itil

Tables installed

Table	Description
BaseLines [sa_baselines]	Storing points in the time defined as baselines for application services.
Business Service User preferences [sa_business_service_user_prefs]	User preferences associated with a specific application service.
Menu Action [sa_context_menu]	Data on configurable menu options for CIs in the application service map.
Hash [sa_hash]	Internal table which contains counters and hashes on various types of updates related to application services.
Entry Point [sa_m2m_service_entry_point]	Maps entry points to application services.
Discovered Service Notification [sa_notification]	Internal table which contains data on notifications between different parts of the software. Mostly used after activating Service Mapping.
Service Group Members [sa_service_group_member]	Maps service groups to application service members.
Business Service Group Responsibilities	Data on users having access to application service groups.

Table	Description
[sa_service_group_responsibilities]	
Checkpoint Attribute Description [checkpoint_attribute_description]	Links between history timeline changes and service model internal entities (checkpoints). Used in lists of history of changes in application service maps.
Application Service [cmdb_ci_service_auto]	Services that can be monitored by the system, which in the base system, includes only application services. If Service Mapping is activated, there can also be records for dynamic CI groups. If Event Management is activated, there can be records for alert groups.
Mapped Application Service [cmdb_ci_service_discovered]	Application service CIs. For each application service, there is a container CI record that models the application service.
Bulk Convert Services [cmdb_convert_bulk_services]	All bulk conversions of business services to application services (current and past), along with the conversion progress which is refreshed every 10 seconds.
Application Services Action Results [csdm_dashboard_action_report_result]	Results for the 'Application Services Missing Data' report in the Application Service Dashboard.
Application Services Types Results [csdm_dashboard_type_report_result]	Results for the 'Application Services by Type' report in the Application Service Dashboard.

Table	Description
Application Services Dashboard Results [csdm_dashboard_reports_result]	Results for the '<Application Servers Databases Hardware Servers> Not in an Application Service' reports in the Application Service Dashboard.
Manual CI Inclusions / Exclusions [svc_manual_ci_exclusions_inclusions.list]	<p>Contains CI classes included or excluded from application services during population of manual or dynamic application services. Service population happens when</p> <ul style="list-style-type: none"> • Manually adding CIs to an application service • Converting a business service to an application service • Creating or updating an application service using APIs • Manually updating an application service with changes from the CMDB <p>In the base system, the following CI classes are excluded:</p> <ul style="list-style-type: none"> • cmdb_ci_file_system • cmdb_ci_network_adapter • cmdb_ci_storage_device • cmdb_ci_disk_partition • cmdb_ci_memory_module • cmdb_ci_ip_address • cmdb_ci_storage_pool_member • dscy_net_base

Table	Description
	<ul style="list-style-type: none">• cmdb_ci_storage_export• cmdb_ci_endpoint• cmdb_ci_translation_rule• cmdb_ci_qualifier• cmdb_ci_application_cluster• cmdb_ci_config_file <p>CLs of any CI class that is not configured for exclusion in this table can be added to application services.</p> <p>This table supports the functionality that was earlier supported using the following deprecated property: sa.mapping.user.manual.citype.bl acklist.</p>

Properties installed

To access application services properties, navigate to **All > Configuration > Application Services > Properties**. The role required for modifying property values, is app_service_admin.

If Service Mapping is deployed, see [Properties installed with Service Mapping](#) for additional application service properties.

Note: To open the System Properties [sys_properties] table, enter sys_properties.list in the navigation filter.

Property	Usage
<p>The sys_id of the default relation type to be added between source and target when adding CI manually to application service</p> <p>sa.it_service.manual_ci_rel_type</p>	<ul style="list-style-type: none"> Type: string Default value: 5599a965c0a8010e00da3b58b113d70e (Depends on::Used by) Learn more: Manually add CIs to an application service
<p>Coefficient of aggregation interval. 0 value means no aggregation is performed on history timeline. The purpose of this property is to decrease number of changes in history timeline by increasing the interval allowed between changes</p> <p>sa.history.aggr_interval_coef</p>	<ul style="list-style-type: none"> Type: integer Default value: 1
<p>A list of comma delimited CI types that are excluded when using the 'Convert to Application Service' and 'Populating Application Service from CIs in the CMDB' operations. Example: cmdb_ci_service,cmdb_ci_endpoint,cmdb_ci_hardware</p> <p>sa.mapping.user.manual.citype.blacklist</p>	<p>This property is deprecated in Tokyo.</p> <p>This exclusion list applies when:</p> <ul style="list-style-type: none"> Manually adding CIs to an application service Converting a business service to an application service Creating or updating an application service using APIs

Property	Usage
	<ul style="list-style-type: none"> Manually updating an application service with changes from the CMDB Type: string Default value: None
Sync Service Mapping operations with Service Modeling sa.service_modeling.use	<ul style="list-style-type: none"> Type: true false Default value: true
Enable limitation of application service maps drawing by number of nodes and edges. sa.map.LIMIT_MAX_GRAPH_SIZE	Limit the number of nodes and edges on application service maps. <ul style="list-style-type: none"> Type: true false Default value: true <p>Setting this property to false may reduce performance in maps of large services.</p>
Maximal number of displayable nodes on application service map. Maps with larger values will not be displayed. sa.map.MAX_NODES_FOR_LAYOUT	The max number of nodes displayed on an application service map. If the number of nodes exceeds the specified number, the map does not appear and an error message appears. <ul style="list-style-type: none"> Type: integer Default value: 5000

Property	Usage
<p>Global flag to allow or disable spanning tree view for maps. true (default) - allows but not forces spanning tree view on maps.</p> <p>sa.map.ALLOW_SPANNING_TREE_VIEW</p>	<p>Enable spanning tree view for application service maps.</p> <ul style="list-style-type: none"> Type: true false Default value: true
<p>Maximal number of displayable edges on application service map before spanning tree view applied.</p> <p>sa.map.MAX_EDGES_FOR_FULL_LAYOUT</p>	<p>The max number of edges displayed on an application service map, before applying spanning tree view.</p> <ul style="list-style-type: none"> Type: integer Default value: 1000
<p>Maximal number of displayable edges on application service map. Maps with larger values will not be displayed.</p> <p>sa.map.MAX_EDGES_FOR_LAYOUT</p>	<p>Max number of edges displayed on an application service map. If the number of edges exceeds the specified number, the map does not appear and an error message appears.</p> <ul style="list-style-type: none"> Type: integer Default value: 100000 <p>Increasing the default value may reduce performance in maps for large services.</p>
<p>Maximal degree of node on application service map for large map mode. Maps with smaller</p>	<ul style="list-style-type: none"> Type: integer Default value: 1000

Property	Usage
<p>degrees will be displayed in regular mode. Maps with larger degrees will apply more edges merging for more compact view.</p> <p>sa.map.LIMIT_GRAPH_DEGREE</p>	<p>Increasing the default value may reduce performance in maps for large services.</p>
<p>Limit of amount of services that displayed on Services Tree on maps. Then this limit reached, Services Tree will be blocked.</p> <p>sa.service_tree.MAX_ITEMS_TO_DISPLAY</p>	<ul style="list-style-type: none"> Type: integer Default value: 7000
<p>Maximal amount of connection properties to be shown at once when connection line selected on service map. If selected line contains more connections than defined here, then properties panel will have notification about cut-off connections.</p> <p>sa.map.max_connections_in_properties_panel</p>	<ul style="list-style-type: none"> Type: integer Default value: 50
<p>Enable grouping of CIs on map.</p> <p>sa.map.enable_auto_grouping</p>	<ul style="list-style-type: none"> Type: true false Default value: true
<p>Minimal number of CIs on a map to apply CI grouping. Relevant only if CI grouping is enabled on</p>	<ul style="list-style-type: none"> Type: integer Default value: 10

Property	Usage
<p>the map. The following CIs are not counted: discovered clusters, internal CIs inside inclusion boxes, entry points, error nodes, host CIs or CIs that are not hosted on other CIs.</p> <p><code>sa.map.min_nodes_for_auto_grouping</code></p>	
<p>Render full labels on CIs on map. Applicable to all CI labels (CI name, host name, cluster label, etc.) Enabling this will disable labels truncation, and labels will most probably overlap with other map elements. Not applicable to network/storage path maps.</p> <p><code>sncCommonMap.RENDER_FULL_LABELS</code></p>	<p>The default value of disabled, means none.</p> <ul style="list-style-type: none"> • Type: choice list • Default value: Disabled • Other possible values: <ul style="list-style-type: none"> • Exported PDF only: pdf • Map and PDF views: all
<p>Maximal width of CI node labels in pixels. Relevant for any kind of labels (CI name, host name, cluster label etc.) This size also modifies horizontal space between CI elements. Applied to map view and exported PDF view. Not applicable to network/storage path maps.</p> <p><code>sncCommonMap.NODE_LABEL_WIDTH</code></p>	<ul style="list-style-type: none"> • Type: integer • Default value: 95 • Other possible values: <ul style="list-style-type: none"> • Min value: 20 • Max value: 1000

Property	Usage
glide.cmdb.csdm.app_service.max_results	<p>Max number of items that are calculated in the '<Application Servers Databases Hardware Servers> Not in an Application Service' report in the Application Service Dashboard.</p> <ul style="list-style-type: none"> Type: integer Default value: 100000 Location: Add to System Properties [sys_properties] table.
sa.service_max_ci_service_population	<p>The maximum number of CI connections added to application services during the following operations: Converting manual services created in Event Management into application services and updating application services with changes from the CMDB.</p> <ul style="list-style-type: none"> Type: integer Default value: 1,000 Location: Add to System Properties [sys_properties] table. <p>Increasing the default value may cause performance issues.</p>
sa.service.population.stop_expansion_under_ci_classes	List of application service CI classes. If an application service belongs to a CI class that extends one of the CI classes in the list, the system does not insert CIs under this application service CI during

Property	Usage
	<p>Manually updating an application service with changes from the CMDB.</p> <ul style="list-style-type: none">• Type: string• Default value: cmdb_ci_service_discovered• Location: System Property [sys_properties] table.

Data Certification

Data Certification manages scheduled and on-demand validations of the configuration management database (CMDB) data.

Information is added to the CMDB by Discovery, by importing from third-party tools, or manually. For regulatory or procedural reasons, information in the CMDB requires checks for accuracy and certification. The person or team responsible for certification can define what information requires verification and a verification schedule. The schedule then generates a checklist for verifying the data. Individuals assigned to certification tasks answer a series of questions to verify the data.

Data certification can be performed against specific fields on specific tables. Based on the certification schedule, certification tasks are automatically created and assigned. For example, you can set up a certification to validate key information fields, such as **Operating System** and **CPU count**, on all Windows servers located in Chicago. You can then assign the tasks to the appropriate team member automatically.

Domain separated systems can use the Data Certification application.

- [Activate Data Certification](#)

Activate the Data Certification plugin to access the application. Activating this plugin also activates the Version Management plugin, which manages certification filter versions.

- **Certification schedules**

A certification schedule defines the information that requires certification and the frequency of execution.

- **Certification tasks**

A certification task represents the work of verifying the data associated with a particular record.

- **Certification elements**

Each element of each record being certified is tracked in its own certification element record.

- **Certification instances**

A certification instance is the collection of certification tasks for one execution of a certification schedule.

- **Certification audit instances**

A certification audit instance is a collection of the certification instances and tasks generated by a single execution of the certification audit definition.

- **Certification audit definition**

A certification audit definition is a collection of certification schedules that can be run at once.

- **Data Certification Overview module**

The Data Certification Overview module displays various data certification-related reports on the Data Certification Console homepage.

- **Data Certification planning**

Initial planning can make the certification process more successful.

- **Data certification performance**

After the certification process has been planned, certification tasks can be performed according to defined schedules.

- [Reassign a certification task](#)

If you have the certification_admin role, you can reassign any certification task in the Work in Progress state. Tasks in Closed Complete, Closed Incomplete, or Cancelled state cannot be reassigned. When a task is reassigned, the current task owner and the new task owner are sent a message.

- [Send certification task reminders](#)

The Certification Task Escalations workflow sends automatic email reminders.

- [Mark a certification task as closed incomplete](#)

Mark a task as closed incomplete if, for example, only some of the elements can be certified.

- [Certification tasks cancellation](#)

Users with the certification_admin role can cancel a certification task in the Work in Progress or Closed Incomplete state.

- [Domain separation and Data Certification](#)

Domain separation is supported in Data Certification processing. Domain separation enables you to separate data, processes, and administrative tasks into logical groupings called domains. You can control several aspects of this separation, including which users can see and access data.

Activate Data Certification

Activate the Data Certification plugin to access the application. Activating this plugin also activates the Version Management plugin, which manages certification filter versions.

Before you begin

Role required: admin

Procedure

1. Navigate to **All > System Applications > All Available Applications > All**.
2. Find the Data Certification plugin using the filter criteria and search bar.

You can search for the plugin by its name or ID. If you cannot find a plugin, you might have to request it from ServiceNow personnel.

3. Select **Install**, and then in the Activate Plugin dialog box, select **Activate**.

Note: When domain separation and delegated admin are enabled in an instance, the administrative user must be in the **global** domain. Otherwise, the following error appears: Application installation is unavailable because another operation is running: Plugin Activation for <plugin name>.

- [Installed With Data Certification](#)

Activating the Data Certification plugin installs the following components.

Related tasks

- [Reassign a certification task](#)
- [Mark a certification task as closed incomplete](#)

Related concepts

- [Certification schedules](#)
- [Certification tasks](#)
- [Certification elements](#)
- [Certification instances](#)
- [Certification audit instances](#)
- [Certification audit definition](#)

- Data Certification planning
- Data certification performance
- Certification tasks cancellation
- Domain separation and Data Certification

Related reference

- [Data Certification Overview module](#)
- [Send certification task reminders](#)

Installed With Data Certification

Activating the Data Certification plugin installs the following components.

Demo data is available with Data Certification. The demo data provides information including filters, schedules, instances, and tasks.

Tables

Data Certification adds the following tables:

Table	Description
Certification Audit Definition [cert_audit_definition]	Stores collections of certification schedules that can be run as a single entity.
Certification Audit Definition Elements [m2m_cert_audit_def_cert_sched]	Lists the certification schedules in each certification audit definition.
Certification Audit Instance [cert_audit_instance]	Stores the certification instances associated with a specific audit definition.
Certification Element [cert_element]	Stores the data elements that are grouped into certification tasks.

Table	Description
Certification Filter [cert_filter]	Stores the data that requires certification using a filtering condition for the certification.
Certification Instance [cert_instance]	Stores a collection of certification tasks representing a single instance of a scheduled certification. This table extends the Audit [cert_audit] table.
Certification Schedule [cert_schedule]	Stores certification for a specific set of information on a specific table, what user or group the tasks are assigned to, and how often this certification is done.
Certification Task [cert_task]	Stores individual certification tasks. Certification Task extends the Task table.

Script Includes

Data Certification adds the following script includes:

Name	Description
CertificationAjax	Provides utilities that enable individual certification elements to be certified, rejected, or reverted.
CertificationTaskCreate	Custom code that extends the standard code for certification tasks.
CertTaskEscalationTimerPercentage	Updates time and percentage complete information for a certification.
CertificationUtilities	Provides utility functions for certification.

Client Scripts

Data Certification adds the following client scripts:

Name	Table	Description
Alert If Boxes Checked	Certification Task [cert_task]	Provides a warning if the certifier attempts to leave a record without certifying the checked elements
Check Table Name	Certification Schedule [cert_schedule]	Updates the table name when a different filter is selected.

UI Policies

Data Certification adds the following UI policies:

Name	Table	Description
Hide next scheduled run	Certification Schedule [cert_schedule]	Hides the Next Scheduled Run field when the schedule is set to run once or on demand only.
<ul style="list-style-type: none">• Hide "run" associated fields when active is set to false• Hide Run When Not Active	Certification Schedule [cert_schedule]	Hides the Run field when Active is set to False.
Make table name read only	Certification Schedule [cert_schedule]	Makes the Table field read-only.

Name	Table	Description
Hide Table field	Certification Element [cert_element]	Hides the Table field on the certification task form.
Make percent complete field read only	Certification Instance [cert_instance]	Makes the Percent complete field read only when the State is Work in Progress, Closed Complete, Closed Incomplete, or Cancelled.
Show Assign to fields	Certification Schedule [cert_schedule]	Shows the Assign To field when the assignment type is User and hides the Assign To field for all other assignment types.
Show Group field	Certification Schedule [cert_schedule]	Shows the Assignment Group field when the assignment type is Group and hides the Assignment Group field for all other assignment types.
Show User field	Certification Schedule [cert_schedule]	Shows the User field when the assignment type is User.
Show Assignment Fields	Certification Schedule [cert_schedule]	Shows the Assign To Empty option when the assignment type is User Field or Group Field.

Business Rules

Data Certification adds the following business rules:

Name	Table	Description
Adjust dates for cert tasks	Certification Instance [cert_instance]	Adjusts dates for tasks belonging to the certification instance when the dates are changed for an active certification.
Cancel Instance	Certification Instance [cert_instance]	Cancels all open certification tasks when an active certification is canceled.
certification audit instance events	Certification Audit Instance [cert_audit_instance]	Sends an inserted event when an active certification audit instance is created. Sends a completed event when an active certification audit instance is marked as complete or incomplete.
certification element events	Certification Element [cert_element]	Sends a failed event when an element of a certification is marked as failed.
certification instance events	Certification Instance [cert_instance]	Sends an inserted event when an instance of a certification is created. Sends a completed event when an instance of a certification is completed.

Name	Table	Description
Certification Instance Rollup	Certification Task [cert_task]	Updates the Percent complete field on the certification instance record.
certification task events	Certification Task [cert_task]	Sends an inserted event when a task is inserted. Sends a completed event when a task is deactivated. Sends a canceled event when a task is canceled.
Certification Task Values	Certification Element [cert_element]	Updates the percent complete of the parent task when a certification element is updated.
Check Certification Audit Progress	Certification Instance [cert_instance]	Updates the completion status of the audit instance as a whole when a certification that is part of an audit is complete.
Clean Certification Views	Certification Instance [cert_instance]	Cleans all related records when a certification instance is deleted.
Copy certification schedule fields	Certification Instance [cert_instance]	Copies changes to the certification schedule to the certification instance.
Merge Certification Tasks	Certification Task [cert_task]	Merges two tasks together when a task is reassigned and

Name	Table	Description
		there is another task for the same instance with the new user.
Prevent delete of Filter with Schedule	Certification Filter [cert_filter]	Prevents the deletion of a filter that is used in a schedule.
Reassign Notification	Certification Task [cert_task]	Sends out a notification to the new and previous assignees when a task is reassigned.
Rollup State	Certification Task [cert_task]	Updates all necessary parent items when task state is changed.
Update audit reference	Certification Task [cert_task]	Makes Data Certification records compatible with Desired State records. This rule makes sure that the Audit field is correctly completed when a record is inserted using Insert and Stay.
Update audit result	Certification Element [cert_element]	Makes Data Certification records compatible with Desired State records for reporting purposes. This rule puts certified values in the Desired value column when an audit is Certified. It also puts actual values in the Discrepancy

Name	Table	Description
		value column when an audit is Failed.
Update follow_on_task & audit references	Certification Element [cert_element]	Makes Data Certification records compatible with Desired State records for reporting purposes. This rule makes certification tasks compatible with follow-on tasks and displays all tasks, regardless of origin.
Update next run time	Certification Schedule [cert_schedule]	Updates the Next scheduled run field when a schedule runs Daily, Weekly, Monthly, or Periodically.
Verify Fields	Certification Schedule [cert_schedule]	Verifies that no field is used in both Display and Certification fields when the fields of a certification schedule are changed.

Formatter

Data Certification adds the following formatter:

Formatter

Name	Description
Certification Task Elements	Enables custom user interface formatting of elements on a certification task. For example, displays the green check mark

Name	Description
	and red exclamation point to use when certifying an element.

Properties

Properties

Name	Table	Description
glide.ui.cert_task_activity.fields	System Properties [sys_properties]	Defines which journal field is the task activity field. Default: work_notes

User Roles

Data Certification adds the following user roles:

User Roles

Role	Contains Roles	Description
certification_admin	certification	Can: <ul style="list-style-type: none">Create and configure certificationsOverride provided answersPerform certification tasks for certification task ownersSend certification task notifications to users and owners at any time

Role	Contains Roles	Description
		<ul style="list-style-type: none"> Cancel or delete certifications in any state
certification_filter_admin	certification	Can create and manage all data certification filters.
certification	none	Can update active or incomplete tasks assigned to them or to groups of which they are a member. Can also update configuration items owned by them or by groups of which they are a member. Receives email notifications when assigned certification tasks.

Events

Data Certification adds the following events. The ServiceNow system uses these events to send email notifications to task owners and managers about changes in certification records.

Name	Description
cert_audit_instance.completed	A certification audit instance has been completed.
cert_audit_instance.inserted	A certification audit instance has been inserted.
cert_element.failed	A certification element has failed certification.

Name	Description
cert_instance.complete	A certification instance has been completed.
cert_instance.inserted	A certification instance has been inserted.
cert_task.cancelled	A certification task has been canceled.
cert_task.completed	A certification task has been completed.
cert_task.escalate	A certification task record has been escalated.
cert_task.inserted	A new certification task has been created.
cert_task.notifications	A certification task notification has been resent to a user.
cert_task.overdue	A certification task is past its specified completion date.
cert_task.reassign	A certification task has been reassigned.
cert_task.warning	A new task escalation point has been reached.

Email Templates

Data Certification adds the following email templates:

Name	Message
certification.task.cancelled	A certification task assigned to you/your group as part of the data certification and management process has been canceled.

Name	Message
certification.task.reminder.inserted	A certification task that has been assigned to you/your group as part of the data certification and management process requires attention.
certification.task.reminder.outstanding	A certification task that has been assigned to you/your group as part of the data certification and management process requires attention.
certification.task.reminder.overdue	A certification task that has been assigned to you/your group as part of the data certification and management process is overdue.

Certification schedules

A certification schedule defines the information that requires certification and the frequency of execution.

At each time interval specified, or on-demand, the certification schedule generates a set of certification tasks based on set conditions. Use the Preview Certification Tasks related link to preview the certification tasks generated from a certification schedule.

Certification schedule

Certification Schedule
Unix Server in South Africa

Certification Schedule		Actions	
Name	Unix Server in South Africa	Active	<input checked="" type="checkbox"/>
Filter	Unix Server in South Africa - 1	Run	Monthly
Table	UNIX Server [cmdb_ci_unix_s...]	Day	1
* Display fields	<input type="button" value="Lock"/>	Time	Hours 00 00 00
* Certification fields	<input type="button" value="Lock"/>	Last run date	2012-10-29 12:22:07
Assignment type	Specific User	Next scheduled run	2012-11-01 00:00:00
* User	Abel Tuter		
* Days to complete	2		
Task description	Ensure that all Unix Servers in South America data center meet minimum reqs		
Instructions	<p>The minimum requirements for Unix Servers are:</p> <ul style="list-style-type: none"> • CPU Speed: 3ghz • Ram: 16gb • CPU Core Count: 4 		
<input type="button" value="Update"/> <input type="button" value="Execute Now"/> <input type="button" value="Delete"/>			

Related tasks

- Activate Data Certification
- Reassign a certification task
- Mark a certification task as closed incomplete

Related concepts

- Certification tasks
- Certification elements
- Certification instances
- Certification audit instances

- Certification audit definition
- Data Certification planning
- Data certification performance
- Certification tasks cancellation
- Domain separation and Data Certification

Related reference

- [Data Certification Overview module](#)
- [Send certification task reminders](#)

Certification tasks

A certification task represents the work of verifying the data associated with a particular record.

Task owners are responsible for performing the certification tasks. Tasks have an associated workflow that sends reminders to the task owner and, if necessary, the manager of the owner at regular intervals.

Certification task

The screenshot shows two ServiceNow pages related to a certification task.

Certification Task:

- Number:** TSK0009072
- Assigned to:** Abel Tuter
- Assignment group:** (empty)
- Complete by:** 2018-11-13 15:04:57
- State:** Closed Complete
- Percent complete:** 100
- Escalation:** Normal
- Short description:** Ensure that all Unix Servers in South America data center meet minimum reqs
- Work notes:** Work notes

Certification data for Certification Task TSK0009072:

	CPU speed (MHz)	RAM (MB)	CPU core count
<input type="checkbox"/>	1,650	8,192	1
<input type="checkbox"/>	1,650	8,192	1
<input type="checkbox"/>	1,650	8,192	1

Note: If the message

Record cannot be certified until the instance is finished creating all certification tasks and elements. Reload the page to try again

appears, it signifies that:

- A large amount of data is present in the cmdb_ci and cmdb_ci_server tables.
- Data certification task processing is not complete (Data Certification jobs are still in process).

As directed, reload the page and wait for the processing to complete.

Clean up invalid elements

Use the **Clean up invalid elements** UI action to query and delete certification elements that reference invalid records. Each certification task has a certification schedule, and each certification schedule has Table and Filter fields. When you use this UI action, it performs the following processing:

1. Collects all available records from Table field in the certification schedule with filters that are available in certification schedule.
2. Collects all certification elements associated with the current certification task.
3. Deletes the certification elements that are no longer available for the data collected in the previous step.
4. After deleting invalid records, it recomputes the certification completion percentage using the following formula:
$$(1 - (\text{number of certification elements pending} / \text{total no of certification elements associated})) * 100;$$
5. If there are no certification elements with a Pending status, it marks the associated certification task as Closed, and deactivates it.
6. If there are remaining certification elements with a Pending status, it activates the associated certification task and changes its status to Work in Progress.

Related tasks

- [Activate Data Certification](#)
- [Reassign a certification task](#)
- [Mark a certification task as closed incomplete](#)

Related concepts

- [Certification schedules](#)
- [Certification elements](#)

- Certification instances
- Certification audit instances
- Certification audit definition
- Data Certification planning
- Data certification performance
- Certification tasks cancellation
- Domain separation and Data Certification

Related reference

- [Data Certification Overview module](#)
- [Send certification task reminders](#)

Certification elements

Each element of each record being certified is tracked in its own certification element record.

Also tracked are the date and time when the element was certified, comments, and the original and certified values of the field. You can view elements on individual certification tasks.

Certification elements

Certification Elements							New	Go to	Certified	Search	
		All	Certified	Document	State	Element	Original value	Certified value	Comment	Certification Task	Audit
		Search	Search	Search	Search	Search	Search	Search	Search	Search	Search
» Audit: Certify groups (8)											
» Audit: Certify servers (160)											
» Audit: IBM Server Schedule (7)											
» Audit: Oracle Database Schedule (5)											
<input type="checkbox"/>	i	2015-08-29 15:10:02	Database: bond_trade_uk	● Certified	change_control			TSK0009071		Oracle Database Schedule	
<input type="checkbox"/>	i	2015-08-29 15:10:02	Database: bond_trade_ny	● Certified	change_control			TSK0009071		Oracle Database Schedule	
<input type="checkbox"/>	i	2015-08-29 15:10:05	Database: SAP ORA01	● Failed	change_control			TSK0009071		Oracle Database Schedule	
<input type="checkbox"/>	i	2015-08-29 15:10:02	Database: NY RAC	● Certified	change_control			TSK0009071		Oracle Database Schedule	
<input type="checkbox"/>	i	2015-08-29 15:10:02	Database: PS ORA01	● Certified	change_control			TSK0009071		Oracle Database Schedule	
» Audit: Unix Server in South Africa (9)											
» Audit: Verify Oracle (5)											
» Audit: Verify User Lenovo Computers (64)											

Related tasks

- Activate Data Certification
- Reassign a certification task
- Mark a certification task as closed incomplete

Related concepts

- Certification schedules
- Certification tasks
- Certification instances
- Certification audit instances
- Certification audit definition
- Data Certification planning

- Data certification performance
- Certification tasks cancellation
- Domain separation and Data Certification

Related reference

- Data Certification Overview module
- Send certification task reminders

Certification instances

A certification instance is the collection of certification tasks for one execution of a certification schedule.

Certification instances

The screenshot shows two stacked tables from the ServiceNow interface. The top table is titled 'Certification Instances' and has columns for Number, Certification Schedule, Percent complete, Created, and Complete by. It shows two rows: 'TSK0009064' (Closed Complete, Bow Ruggeri, 100%) and 'TSK0009066' (Closed Incomplete, Don Goodliffe, 0%). The bottom table is titled 'Certification Tasks' and has columns for Search, Search, Search, Search, and Search. It lists 15 tasks across five rows, each with a progress bar indicating completion percentage. The tasks include 'Certify servers', 'IBM Server Schedule', 'Oracle Database Schedule', 'Unix Server in South Africa', 'Verify Oracle', and 'Verify User Lenovo Computers'. The first two rows have 3 tasks each, and the last three rows have 3 tasks each.

Number	Certification Schedule	Percent complete	Created	Complete by
TSK0009064	Bow Ruggeri	100%		
TSK0009066	Don Goodliffe	0%		

Search	Search	Search	Search	Search
CRTINST0009010	Certify servers	3.03%	2013-03-26 16:20:30	2013-06-16 15:04:27
CRTINST0009011	IBM Server Schedule	0%	2012-03-11 16:04:37	2012-12-16 14:04:37
CRTINST0009012	Oracle Database Schedule	100%	2013-12-30 14:04:37	2014-03-20 16:04:37
CRTINST0009013	Unix Server in South Africa	100%	2013-03-11 16:19:21	2014-01-22 15:04:57
CRTINST0009014	Verify Oracle	0%	2013-10-10 15:04:57	2014-02-16 15:04:57
CRTINST0009015	Verify User Lenovo Computers	4.69%	2013-02-08 15:05:07	2013-04-11 16:05:07

Related tasks

- Activate Data Certification
- Reassign a certification task

- Mark a certification task as closed incomplete

Related concepts

- Certification schedules
- Certification tasks
- Certification elements
- Certification audit instances
- Certification audit definition
- Data Certification planning
- Data certification performance
- Certification tasks cancellation
- Domain separation and Data Certification

Related reference

- Data Certification Overview module
- Send certification task reminders

Certification audit instances

A certification audit instance is a collection of the certification instances and tasks generated by a single execution of the certification audit definition.

Certification audit instance

Certification Audit Instances				
Number		Certification Audit Definition	Created	Complete by
Search	Search	Search	Search	Search
<input type="checkbox"/>	CRTAUD00000001	Certify Servers and Groups	2018-10-30 08:09:36	2018-11-13 07:09:36

Related tasks

- [Activate Data Certification](#)
- [Reassign a certification task](#)
- [Mark a certification task as closed incomplete](#)

Related concepts

- [Certification schedules](#)
- [Certification tasks](#)
- [Certification elements](#)
- [Certification instances](#)
- [Certification audit definition](#)
- [Data Certification planning](#)
- [Data certification performance](#)
- [Certification tasks cancellation](#)
- [Domain separation and Data Certification](#)

Related reference

- [Data Certification Overview module](#)
- [Send certification task reminders](#)

Certification audit definition

A certification audit definition is a collection of certification schedules that can be run at once.

Certification audit definition

The screenshot shows the 'Certify Servers and Groups' certification audit definition. It includes fields for Name (Certify Servers and Groups), Days to complete (14), and Description (Annual certification of Servers and Groups). Below the form are 'Related Links' and a 'Create Certification Audit Instance' button. A 'Certification Schedules' list view is displayed, showing two entries: 'Certify groups' (sys_user_group) and 'Certify servers' (cmdb_ci_server).

Element	Type	Value	Description
Name	Text	Certify Servers and Groups	
Days to complete	Text	14	
Description	Text	Annual certification of Servers and Groups	

Related Links

[Create Certification Audit Instance](#)

Certification Schedules (2)

Element	Type	Value	Description
Certify groups	sys_user_group	cost_center,email,type,default_assignee	
Certify servers	cmdb_ci_server	classification,cost_center,owned_by,support_group	

Related tasks

- [Activate Data Certification](#)
- [Reassign a certification task](#)
- [Mark a certification task as closed incomplete](#)

Related concepts

- [Certification schedules](#)
- [Certification tasks](#)
- [Certification elements](#)
- [Certification instances](#)
- [Certification audit instances](#)

- Data Certification planning
- Data certification performance
- Certification tasks cancellation
- Domain separation and Data Certification

Related reference

- [Data Certification Overview module](#)
- [Send certification task reminders](#)

Data Certification Overview module

The Data Certification Overview module displays various data certification-related reports on the Data Certification Console homepage.

The Overview module is a type of homepage.

The different levels of access are:

Access levels per role

Role	Access
certification	View (view overview page and refresh reports)
certification_admin	<ul style="list-style-type: none">• View (view overview page and refresh reports)• Customize (refresh, add, delete, and rearrange reports) View, customize

Role	Access
admin	<ul style="list-style-type: none">View (view overview page and refresh reports)Customize (refresh, add, delete, and rearrange reports)Edit (can edit reports)

Data Certification Overview Module

The Overview module includes the following reports:

Data Certification Overview Module Description

Report	Description	Table
30/60/90 Day Aging	Groups tasks by the number of days (30, 60, 90, and 90 and over) since the task was opened.	Certification Task
Certification Instances	Lists all certification instances.	Certification Instance
Certification Progress Report	Groups tasks by task owner, indicating task progress as a percentage.	Certification Task
Certification Task Completed Report	Groups tasks by task owner, indicating tasks that are complete.	Certification Task
Exceptions To Date	Lists all task elements that have comments added and a state of Failed or In Progress.	Certification Element

Report	Description	Table
Functional Roll Up	Lists the managers that have groups with assigned certification tasks. The report is a horizontal bar chart, grouped by status, with each bar representing a manager of an assignment group.	Certification Task
Hierarchical Roll Up	Shows the managers that have employees with assigned certification tasks (task owners). The report is a horizontal bar chart, grouped by status, with each bar representing a manager of a task owner (identified in the Assigned to field).	Certification Task
Upcoming Schedules	Lists all schedules that are scheduled to run within the next 30 days.	Certification Schedule

- Use the Data Certification Overview module

View the status of data certification tasks.

Related tasks

- [Activate Data Certification](#)
- [Reassign a certification task](#)
- [Mark a certification task as closed incomplete](#)

- Use the Data Certification Overview module

Related concepts

- Certification schedules
- Certification tasks
- Certification elements
- Certification instances
- Certification audit instances
- Certification audit definition
- Data Certification planning
- Data certification performance
- Certification tasks cancellation
- Domain separation and Data Certification

Related reference

- Send certification task reminders

Use the Data Certification Overview module

View the status of data certification tasks.

Before you begin

Role required: admin

Procedure

1. Navigate to **All > Data Certification > Overview**.
2. Click elements within the reports to obtain more information.

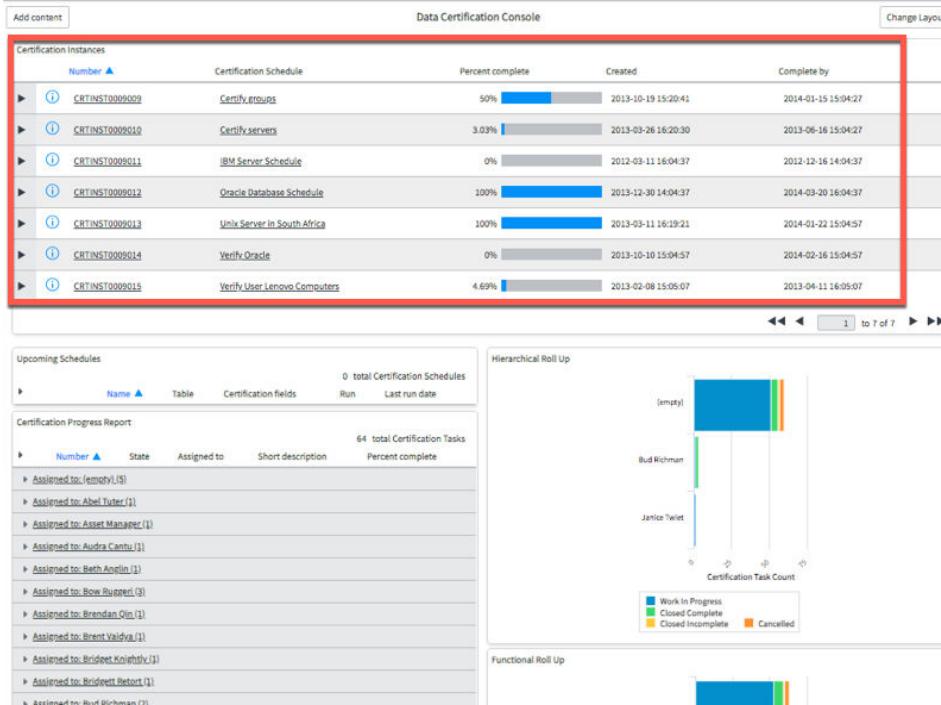
For example, click any of the colored bars in the **Functional Roll Up** bar chart and detailed information replaces the Data Certification Console screen.

3. Update some fields directly on the overview page.

Example

For example, in the red box on the image shown, a certification schedule is being updated in the certification instances report.

Data certification overview module



Related reference

- [Data Certification Overview module](#)

Data Certification planning

Initial planning can make the certification process more successful.

By defining certification schedules and certification audit definitions, users with the certification_admin role establish when certifications are performed, who performs it, and what data must be certified.

Required Roles

Users with the certification_admin role can view filter versions. These users can create, update, and delete filters, if they have the proper access to necessary tables. In the base ServiceNow system, certification_admin users have limited system rights and do not have access to all the tables required for creating a filter. When assigning compliance resources, make sure to grant additional roles to the certification_admin user as needed. For example, this user requires roles that grant access to these tables:

- Company [`core_company`]
- Cost Center [`cmmn_cost_center`]
- Schedule [`cmmn_schedule`]

Planning Data Certification

Planning the data certification process requires defining:

- The certification schedule defines certification for a particular set of information on a particular table. It also generates certification tasks to perform that certification. One certification task is generated per task owner and a certification instance record groups the tasks.
- The optional certification audit definition groups some certification schedules to be performed together and generates certification audit instances to perform them.

The following questions require answers for each certification schedule:

- What information requires certification?
- When is the due date for certification?
- Who must perform the certification?
- [Create a certification filter](#)

A filter is a subset of configuration items from any ServiceNow table that is created with a standard condition builder.

- [Define a certification schedule](#)

A certification schedule specifies the fields to display, the fields that require certification, certification task assignments, completion requirements for task owners, frequency of schedule, and detailed instructions.

- [Preview a certification task](#)

Previewing certification tasks saves any changes to the Certification Schedule form and displays the tasks that are created when you execute the certification schedule.

- [Use a certification schedule notification](#)

After you define a certification schedule, the system automatically sends notifications to specific users based on the information in the schedule.

- [Define and create a certification audit](#)

A certification audit is a collection of certification schedules that can be run as a single entity.

- [Track a certification audit instance](#)

You can view a list of all certification audit instances at any time.

Related tasks

- [Activate Data Certification](#)
- [Reassign a certification task](#)
- [Mark a certification task as closed incomplete](#)

Related concepts

- [Certification schedules](#)
- [Certification tasks](#)

- Certification elements
- Certification instances
- Certification audit instances
- Certification audit definition
- Data certification performance
- Certification tasks cancellation
- Domain separation and Data Certification

Related reference

- [Data Certification Overview module](#)
- [Send certification task reminders](#)

Create a certification filter

A filter is a subset of configuration items from any ServiceNow table that is created with a standard condition builder.

Before you begin

Role required: admin

About this task

An example is a filter that selects all UNIX servers in the Australian data center.

With filters, you can:

- Create multiple versions of a filter and then select the version you want to use.
- Use one filter on multiple certification schedules.
- View the number of records that match your filter as you create the conditions.

Note: Be sure to create certification filters before creating certification schedules.

Procedure

1. Navigate to **All > Data Certification > Certification Filters**.

2. Click **New**.

3. Fill in the form (see table).

4. Click **Submit**.

This action saves the filter as version 1.

Certification filter V1

5. To create another filter version, modify the filter conditions and click **Update**.

The system saves the new filter and increments the version number.
Certification filter V2

By default, the Certification Filters list shows only the current version of each filter. To see all filter versions, click **All** in the breadcrumbs.

Certification Filter List

The screenshot shows a list of certification filters. The first two rows are inactive versions (version 1). The third row is the active version (version 2), which is highlighted with a red border. The fourth row is another version of the same filter (version 1).

	Name	Description	Table	Version
<input type="checkbox"/>	Unix Servers	Only Unix servers	Server [cmdb_ci_server]	1
<input type="checkbox"/>	Unix Server in South Africa	Only Unix Servers in South Africa	UNIX Server [cmdb_ci_unix_server]	1
<input type="checkbox"/>	Oracle Database	Only Oracle Databases	Database [cmdb_ci_database]	1
<input type="checkbox"/>	Linux Servers	Dell Linux servers	Linux Server [cmdb_ci_linux_server]	2
<input type="checkbox"/>	Linux Servers	Dell Linux servers	Linux Server [cmdb_ci_linux_server]	1
<input type="checkbox"/>	IBM Servers	Only IBM Servers	Server [cmdb_ci_server]	1
<input type="checkbox"/>	Data Center Zones	All data center zones	Data Center Zone [cmdb_ci_zone]	1

- To make an inactive filter the current version, open the inactive filter and click **Revert**.

Certification filter revert

The screenshot shows the revert page for the 'Linux Servers' filter. It displays the filter's details: Number (CFLR0010001), Name (Linux Servers), Description (Dell Linux servers), Table (Linux Server [cmdb_ci_linux...]), Version (1), and Filter condition (0 records match condition). Below these fields are buttons for 'Revert' and 'Delete'. A red box highlights the 'Revert' button with the text 'Reverts an inactive version to the current version'.

This action creates a new, active version of the filter and makes all previous versions inactive.

- To delete a single filter version, open that version record and click **Delete**.

8. To delete inactive versions of a filter, click **Delete inactive versions** under **Related Links** in that filter record.

You cannot delete a filter that is used in a schedule definition. The system displays a warning and the filter is not deleted.

Creating certification filters

Field	Description
Name	[Required] Filter name.
Description	[Optional] Brief description of the filter.
Number	[Read-only] Automatically assigned filter identification number.
Table	Table containing the records to be filtered. Use of the Database View [sys_db_view] table is limited by version.
Active	Control to make the filter available for use from the Filter field on the Certification Schedule form.
Version	Current version of this filter. Any significant changes to the filter make the current version inactive. The system copies the updated filter, marks it as active, and increments the version number. The system saves all versions of the filter and makes them available to users. More than one version of a filter can be marked active.
Filter condition	Field, operator, and value to create the condition. The available options depend on the

Field	Description
	<p>table selected. You can view the number of records that match the filter by clicking the refresh icon.</p> <p>Refresh Conditions </p> <p>If the filter does not match any records, the system marks the certification instance as Closed Complete, with the Percent complete value set to 100%.</p>

Related concepts

- [Data Certification planning](#)

Define a certification schedule

A certification schedule specifies the fields to display, the fields that require certification, certification task assignments, completion requirements for task owners, frequency of schedule, and detailed instructions.

Before you begin

Role required: admin

About this task

Use the preview option to see what tasks are created before saving the schedule. If the tasks are not what you want, edit the schedule and preview the tasks again. The system creates certification tasks automatically when it executes a schedule.

To schedule a certification:

Procedure

1. Navigate to **All > Data Certification > Schedule Definitions**.
2. Click **New**
3. Fill in the fields (see table).
4. Click **Submit**.

Defining A Certification Schedule

Field	Description
Name	A schedule name.
Filter	A certification filter for this schedule.
Table	[Read-only] The table holding the records to be certified. To change the table name, select a different Filter or create a new Filter .
Display fields	The fields displayed in the Certification Task list to provide context. These do not require certification themselves. For example, although users are not required to certify the Name field of a record, it displays so that users know what record they are certifying.
Certification fields	The fields to certify on this certification schedule.
Assignment type	A choice list to select how the certification schedule assigns the certification tasks. <ul style="list-style-type: none">• User Field: Select a user reference field on the

Field	Description
	<p>table being certified. As an example, select the user named in the Managed by field to identify the user who performs the task. This selection displays the Assign to and Assign to empty fields. If the reference field on the record is empty, the value in the Assign to empty field is used.</p> <ul style="list-style-type: none">• Specific User: Select a specific user to perform the tasks. This selection displays the User field.• Group Field: Select a group reference field on the table being certified. As an example, select the Support group field to identify the user who performs the task. This selection displays the Assign to group and Assign to empty fields. All members of the group from the reference field on the record are assigned to the tasks. If the reference field on the record is empty, the value in the Assign to empty field is used.• Specific Group: Select a specific group to perform the tasks. This selection displays the Group field. All members of the named group are assigned to the tasks.

Field	Description
User	<p>This field appears when:</p> <ul style="list-style-type: none"> • Assignment type is Specific User. This system assigns this user to all certification tasks for this schedule. • The Assign to empty field is set to Create Assigned Task, and you have selected User Field as the assignment type. The system assigns this user to certification tasks containing unassigned records. <p>You can only select users with the certification role.</p>
Assign to group	<p>The group field that defines the group assigned to the certification tasks. This field is available only when the Assignment type is Group Field.</p>
Group	<p>The specific group to which certification tasks are assigned for this schedule. This field is available only when the Assignment type is Specific Group</p>
Assign to	<p>The user field that defines which user is assigned to the certification task. This field is available only when the Assignment type is User Field.</p>
Assign to empty	<p>The behavior to use if the field selected in Assign to or Assign to group is blank on the record</p>

Field	Description
	<p>being certified. For example, if a task must be assigned to a manager, but no manager is identified, the value in this field determines what happens. This field appears only when the Assignment type is User Field or Group Field. The possible selections are:</p> <ul style="list-style-type: none"> • Do Not Create Task: No task is created when the Assign to or Assign to group field is empty. • Create Unassigned Task: Create a task, but do not assign it to any user or group. The task can be manually assigned later. • Create Assigned Task: Create a task and assign it to the user or group specified. If you selected an assignment type of User Field, the User field is available. If you selected the Group Field type, the Group field is available. <p>The schedule automatically creates certification tasks for all records that do have "Assign to" populated, regardless of which selection you make for "Assign to empty."</p>
Days to complete	[Required] The number of days that task owners have to complete the certification tasks.

Field	Description
	When the certification schedule is part of a certification audit definition, the Days to Complete audit definition value overrides the value set for the certification schedule.
Active	Check box to activate this certification schedule, generating certification tasks at the scheduled date and time. Clear this check box to hide scheduling fields on the form (except Last run date) and not generate certification tasks.
Run	<p>How often to run the schedule that generates certification tasks:</p> <ul style="list-style-type: none"> • Daily • Weekly • Monthly • Periodically • Once • On Demand
Day	<p>When Run is Weekly, the day of the week when the schedule runs and generates certification tasks.</p> <p>When Run is Monthly, the day of the month the schedule runs and generates certification tasks. If the day is 29, 30 or 31, the</p>

Field	Description
	certification runs on the last day of the month for shorter months.
Repeat Interval	When Run is Periodically , the frequency that the schedule runs to generate certification tasks, entered in time, days, or both. For example, set Days to 10 and Hours to 14:00:00 to run the schedule and generate certification tasks every 10 days at 14:00.
Starting	When Run is Periodically or Once , the date and time the schedule runs and generates certification tasks.
Time	When Run is Daily , Weekly , Monthly , or Once , the time of day, on a 24-hour clock, the schedule runs and generates certification tasks.
Last run date	[Read-only] The date and time that the schedule ran last, either on its regular schedule or manually, and generated certification tasks.
Next scheduled run	[Read-only] The next date and time the schedule runs and generates certification tasks.
Task Description	A description to add to the Short Description field of the certification task.

Field	Description
Instructions	An HTML field for providing instructions to the user or group performing the certification.

Related concepts

- [Data Certification planning](#)

Preview a certification task

Previewing certification tasks saves any changes to the Certification Schedule form and displays the tasks that are created when you execute the certification schedule.

Before you begin

Role required: admin

About this task

Previewing tasks is especially useful if you want to test different combinations of options in the **Assignment type**, **Assign to**, and **Assign to empty** fields.

Procedure

1. Navigate to **All > Data Certification > Schedules > Schedule Definitions**.
2. Click a certification schedule **Name**.
3. In **Related Links**, click **Preview Certification Tasks**.
The tasks to be created appear at the top of the screen.

Certification info message

 0 Servers

One Certification Task would be created unassigned to certify 31 Servers

One Certification Task would be created for Bow Ruggeri to certify 7 Servers

One Certification Task would be created for Bud Richman to certify 1 Server

One Certification Task would be created for David Loo to certify 1 Server



Related tasks

- Define a certification schedule

Related concepts

- Data Certification planning

Use a certification schedule notification

After you define a certification schedule, the system automatically sends notifications to specific users based on the information in the schedule.

Before you begin

Role required: admin

About this task

The following notifications are sent automatically:

Certification Schedule Notifications

Time elapsed to end date	Email template name	Notification message is sent to
0% (when task is created)	certification.task.reminder.inserted	Task owner or assignment group, if specified
50%	certification.task.reminder.outstanding	Task owner or assignment group, if specified

Time elapsed to end date	Email template name	Notification message is sent to
75%	certification.task.reminder.outstanding	Task owner, assignment group, if specified, and manager of the task owner, if specified
95%	certification.task.reminder.outstanding	Task owner, assignment group, if specified, and manager of the task owner, if specified
100%	certification.task.reminder.overdue	Task owner, assignment group, if specified, and manager of the task owner, if specified

The email templates used in the notifications can be edited, for example, to change the email message text.

Executing a Certification Schedule

Executing a certification schedule generates certification tasks based on the schedule.

Procedure

1. Navigate to **All > Data Certification > Schedules > Schedule Definitions.**

2. Click a certification schedule **Name**.

3. Click **Execute Now**.

The related lists **Certification Instances** and **Certification Tasks** display the instances or tasks generated by the schedule. The amount of time it takes to generate all certification tasks depends on the size of the table selected and how many fields require certification.

Execute certification schedule

Certification Instances (2)		Certification Tasks (4)		
		Number	State	Assigned to
Certification Schedule = Certify servers				
<input type="checkbox"/>		TSK0009065	Work In Progress	0%
<input type="checkbox"/>		TSK0009067	Cancelled	Bow Ruggeri
<input type="checkbox"/>		TSK0009068	Work In Progress	Bud Richman
<input type="checkbox"/>		TSK0009069	Closed Complete	David Loo

Define and create a certification audit

A certification audit is a collection of certification schedules that can be run as a single entity.

Before you begin

Role required: admin

About this task

Certification audits can be useful when there are multiple certification schedules. After creating a certification audit definition, you can generate a certification audit instance. The certification audit instance is a collection of the certification instances and tasks generated by a single execution of the certification audit definition.

Procedure

1. Navigate to **All > Data Certification > Audits > Audit Definitions**.
2. Click **New**.
3. Fill in the fields (see table).
4. Right-click the header bar and select **Save**.
5. In the **Certification Schedules** related list, click **Edit**.

6. In the **Collection** list on the left, select one or more schedules and click **Add**.

7. Click **Save**.

8. In **Related Links**, click **Create Certification Audit Instance**.

The system generates an audit instance based on the certification schedules selected. All audit instances based on this audit definition are listed in the **Certification Audit Instances** related list.

Defining and Creating a Certification Audit

Field	Description
Name	The name of the audit definition.
Days to Complete	The number of days that task owners have to complete the certification tasks created by this audit definition. Overrides the identical field on the certification schedule.
Description	A short description of the intended audit.

Related concepts

- [Data Certification planning](#)

Track a certification audit instance

You can view a list of all certification audit instances at any time.

Before you begin

Role required: admin

Procedure

1. Navigate to **All > Data Certification > Audits > Audit Instances**.
2. View the **Certification Instances** related list.

The list contains each of the associated instances generated as part of the audit.

Related concepts

- [Data Certification planning](#)

Data certification performance

After the certification process has been planned, certification tasks can be performed according to defined schedules.

Users with the certification role can perform certification tasks. The certification tasks can be tracked as part of certification instances.

- [View and resolve certification tasks](#)

After you execute a certification schedule manually or at a scheduled time, the ServiceNow system performs certain actions.

- [Certify an element](#)

The Certification Task form contains a list of all elements to be certified.

- [Export the certification list](#)

Users with the certification_admin role can export the certifications list and save the list in Excel, CSV, XML, or PDF format. This list is useful when you have a long list of certification elements or if many different users are assigned to certify elements on a single certification schedule.

- [Reset certifications](#)

You cannot reset any element after all elements are certified.

- [Track a task with a certification instance](#)

The Certification Tasks related list on the certification instance record provides information about associated tasks.

- [Cancel a certification instance](#)

Users with the certification_admin role can cancel a certification instance.

- [Track a certification task](#)

Use the certification task state to track the progress of a task.

- [Escalate a certification task](#)

Users with the certification_admin role can escalate a task in the Work in Progress state. To escalate a task, the task owner identified in the Assigned to field on the task record must have an associated manager.

- [Escalate a certification task from the certification task list](#)

Escalate a certification task to notify the manager of the current task owner.

Related tasks

- [Activate Data Certification](#)
- [Reassign a certification task](#)
- [Mark a certification task as closed incomplete](#)
- [Define a certification schedule](#)

Related concepts

- [Certification schedules](#)
- [Certification tasks](#)
- [Certification elements](#)
- [Certification instances](#)
- [Certification audit instances](#)
- [Certification audit definition](#)
- [Data Certification planning](#)
- [Certification tasks cancellation](#)

- Domain separation and Data Certification
- Data Certification planning

Related reference

- [Data Certification Overview module](#)
- [Send certification task reminders](#)
- [Installed With Data Certification](#)

View and resolve certification tasks

After you execute a certification schedule manually or at a scheduled time, the ServiceNow system performs certain actions.

- Creates certification tasks for any records that meet the filter requirements in the specified table, like tasks from the Configuration Item [cmdb_ci] table.
- Assigns the new tasks to the user or group identified in one of these [certification schedule](#) fields:
 - Assign to
 - User
 - Assign to group
 - Group
- Places the new tasks in the Work in Progress state.
- Adds the certification schedule Short description and Assigned to values to the corresponding fields on the certification task record.
- Adds the certification schedule Days to complete and Complete by date fields to the certification task record, based on when the task is created.

Note: If the certification filter does not match any CIs, the system sets the State to Closed Complete and the Percent complete to 100.

To view tasks assigned to you, navigate to **Data Certification > Tasks > My Tasks**. To resolve tasks assigned to you, see [Certify an element](#). For more information about purpose and usage of certification tasks, see [certification tasks](#).

The following information is tracked on the certification task record:

Certification task record

Field	Description
Number	An identification number for the certification task.
Assigned to	The user responsible for certifying the data.
Assignment group	The group responsible for certifying the data.
Complete by	[Read-only] A date field containing a deadline for the task. This field is automatically filled in based on the Days to Complete field on the certification schedule.
State	[Read-only] The current state of the certification task. The selections are: Work in Progress, Closed Incomplete, Closed Complete, and Cancelled.
Percent complete	The task progress as a percentage. This field is read-only when a task is in a Closed Incomplete, Closed Complete, or Cancelled state.
Escalation	[Read-only] The escalation level of the task. When 0–49% of the time to Complete By has elapsed, this field is set to Normal. At 50%, this field changes to Moderate and an email reminder is sent to the task.

Field	Description
	owner. At 75%, this field changes to High and an email reminder are sent to the task owner and the manager of the task owner. At 95%, this field remains set to High, but a second email reminder is sent to the task owner and manager.
Short Description	A short description of the task. This field is automatically filled in with the text from the certification schedule of the Task description field.
Work notes	Information about work performed on the certification.

Certify an element

The Certification Task form contains a list of all elements to be certified.

Before you begin

Role required: admin

About this task

Note: After you certify all the elements in a task, no elements can be reverted.

Procedure

1. Navigate to **All > Data Certification > Tasks > My Tasks**.
2. Open a certification task with a State of Work in Progress.

3. In the upper right corner of the list, select records that require certification for this task or all records that are part of this certification task.

Certification list 3

The screenshot shows a list of certification requirements. At the top right is a button labeled "Show All Records". Below it is a navigation bar with arrows and the text "1 to 20 of 31". The main list has columns for "Certified" (checkbox), "Owned by" (checkbox), and "Support group" (checkbox). There are also empty checkboxes below the first row.

4. Select the check box beside a certification element.

5. In Optional comment for checked elements, above the list, enter information that would be useful to others.

Certification list

The screenshot shows a list of certification requirements. At the top, there is a header "Certifications required for Certification Task TSK0009065 (toggle help)". Below it is a search bar with "Servers" and "New" buttons. A red box highlights the first item in the list: "ApplicationServerPeopleSoft" with a checked checkbox. The list also includes "AS400" with an unchecked checkbox. The interface includes filters for "All" and "Name" (sorted ascending).

6. Do:

- Click the green check mark to certify the element.
 - Click the red exclamation point to fail the element.
7. To see the certified or failed element, set the view to Show All Records.

A green check mark or red exclamation mark appears beside the element.

8. Point to an icon to see any certification comments.
9. Ensure that all elements have the correct certification, either accepted or rejected.

After you certify all elements, no elements can be [reverted](#). When all elements of a certification task are certified or rejected, the task State changes to Closed Complete.

- [View an audit result](#)

View audit results after you certify the elements.

View audit results after you certify the elements.

Before you begin

Role required: admin

Procedure

1. Navigate to **All > Data Certification > Schedules > Audit Results**.

The list of data certification audit results appears, grouped by certification instances. Certified configuration items show the Original value only. Failed CIs contain the Certified value and the Original value.

2. Click the links in the list to open any of the related records.

Data cert audit results

Certification Elements		New	Go to	Certified	Search	<< < > >>						1 to 8 of 8
All		Certified	Document	State	Element	Original value	Certified value	Comment	Certification Task	Audit	418 total Certification Elements	
▶ Audit: Certify groups (8)												
▶ Audit: Certify servers (160)												
▶ Audit: Certify servers (160)												
▶ Audit: IBM Server Schedule (7)												
▶ Audit: Oracle Database Schedule (5)												
<input type="checkbox"/> 2015-08-29 15:10:02 Database: bond_trade_uk	●	Certified		change_control			TSK0009071		Oracle Database Schedule			
<input type="checkbox"/> 2015-08-29 15:10:02 Database: bond_trade_my	●	Certified		change_control			TSK0009071		Oracle Database Schedule			
<input type="checkbox"/> 2015-08-29 15:10:06 Database: SAP ORAO1	●	Failed		change_control			TSK0009071		Oracle Database Schedule			
<input type="checkbox"/> 2015-08-29 15:10:02 Database: NY RAC	●	Certified		change_control			TSK0009071		Oracle Database Schedule			
<input type="checkbox"/> 2015-08-29 15:10:02 Database: PS_ORAO1	●	Certified		change_control			TSK0009071		Oracle Database Schedule			
▶ Audit: Unix Server in South Africa (9)												
▶ Audit: Verify Oracle (5)												
▶ Audit: Verify User Lenovo Computers (64)												

Export the certification list

Users with the certification_admin role can export the certifications list and save the list in Excel, CSV, XML, or PDF format. This list is useful when you have a long list of certification elements or if many different users are assigned to certify elements on a single certification schedule.

Before you begin

Role required: admin

About this task

For general information and common export steps, see [List export](#).

Procedure

1. Navigate to **All > Data Certification > Tasks > All Tasks**
2. Open a task.
3. Open any column context menu in the certification data list and complete the export.

Reset certifications

You cannot reset any element after all elements are certified.

- To reset individual certifications, right-click the element in the certification list and select **Revert Certification**.
- To reset the entire task to its starting point, click the **Reset all Certifications to Pending** related link.

Track a task with a certification instance

The Certification Tasks related list on the certification instance record provides information about associated tasks.

Before you begin

Role required: admin

About this task

The State field on the certification instance record is read-only and is based on the cumulative states of the certification tasks associated with the instance. The Percent complete column allows users with the certification_admin role to track task progress quickly. For more information, see [Track Certification Tasks](#).

To track a certification instance:

Procedure

1. Navigate to **All > Data Certification > Schedules > Instances**.
2. Click a certification instance Number.
3. View and edit the following fields as necessary.

Certification instance

Field	Description
Number	[Read-only] Automatically generated identification number for the instance.
Certification Schedule	The certification schedule used to create the certification instance.
State	[Read-only] Current state of the certification instance: Work in Progress, Complete, Closed Incomplete, or Cancelled. For more information, see Track Certification Tasks .
Created	[Read-only] Date and time the certification instance was created. Date is filled in automatically when the Execute Now button clicks the associated certification schedule.
Complete by	[Required] Date and time when the certification instance must be completed. The system updates this field when it executes the schedule, using the deadline specified on the instance. All certification tasks associated with the certification instance must be marked Complete, Closed Incomplete, or Cancelled before the instance is complete.
Percent complete	Percentage of the instance that has reached the Closed Complete state. This field is automatically filled in based

Field	Description
	on the Percent Complete fields on the associated certification tasks.
Task Description	Information about the certification instance. This field automatically displays the text from the Task description field of the associated certification schedule.
Instructions	Field for providing instructions to the user or group performing the certification. This field is automatically filled in with information from the Instructions field on the associated certification schedule.

Cancel a certification instance

Users with the certification_admin role can cancel a certification instance.

Before you begin

Role required: admin

About this task

The instance must have a State of Work in Progress. Canceling a certification instance:

- Changes the certification instance State to Cancelled.
- Changes all associated Work in Progress certification tasks to Cancelled.

To cancel a certification instance:

Procedure

1. Navigate to **All > Data Certification > Schedules > Instances**.
2. Click a certification instance Number.
3. Click **Cancel**.

Track a certification task

Use the certification task state to track the progress of a task.

Before you begin

Role required: admin

About this task

The available task states are Work in Progress, Closed Complete, Closed Incomplete, and Cancelled.

When the state of a certification task changes, the certification instance state also changes in the following cases:

- If any certification task is in Work in Progress state, the certification instance is placed in Work in Progress state.
- If all certification tasks are in Cancelled state, the certification instance is placed in Cancelled state.
- If all certification tasks are in Cancelled or Closed Complete state, the instance is placed in a Closed Complete state. For example, if three certification tasks are Cancelled, and one task is Closed Complete, the instance state is changed to Closed Complete.
- When one certification task is Closed Incomplete and the remainder of the tasks are Cancelled or Closed Complete, the instance is placed in Closed Incomplete.

To view the state of certification tasks:

Procedure

1. Navigate to **All > Data Certification > Tasks** and select **My Tasks** or **All Tasks**.
2. View the State column for each task.

Escalate a certification task

Users with the certification_admin role can escalate a task in the Work in Progress state. To escalate a task, the task owner identified in the Assigned to field on the task record must have an associated manager.

Before you begin

Role required: admin

About this task

Personalize the User form to see the Manager field.

Escalating a task:

- Sends an email message to the task owner and the manager of the task owner stating that the task has been escalated.
- Sets the manager as the new task owner.

The event that triggers the escalation is named cert_task.escalate and the email notification is named Escalation Notification. To edit the text of the email message that is sent, edit the Escalation Notification email notification directly.

For more information, see [Email notifications](#).

To escalate a certification task from the Certification Task form:

Procedure

1. Navigate to **All > Data Certification > Tasks > All Tasks**.
2. Click a certification task Number.

3. Click **Escalate**. If the Escalate button is not available, the user in the Assigned to field does not have an associated Manager.

Escalate a certification task from the certification task list

Escalate a certification task to notify the manager of the current task owner.

Before you begin

Role required: admin

To escalate a task, the task owner identified in the Assigned to field on the task record must have an associated manager.

Procedure

1. Navigate to **All > Data Certification > Tasks > All Tasks**.
2. Select the check box to the left of a certification task Number. Multiple check boxes can be selected.
3. From the Actions on Selected Rows menu below the list, select **Escalate**. If the Escalate button is not available, the user in the Assigned to field does not have an associated Manager. Select multiple tasks from the list. The menu option shows how many tasks are not eligible for escalation, such as Escalate (4 of 6).

Reassign a certification task

If you have the certification_admin role, you can reassign any certification task in the Work in Progress state. Tasks in Closed Complete, Closed Incomplete, or Cancelled state cannot be reassigned. When a task is reassigned, the current task owner and the new task owner are sent a message.

Before you begin

Role required: certification_admin

About this task

The event associated with the reassignment is named cert_task.reassign and the email notification is named Certification Task Reassignment. To edit the text of the email message that is sent, edit the Certification Task Reassignment email notification directly.

For more information, see [Email notifications](#).

To reassign a certification task:

Procedure

1. Navigate to **All > Data Certification > Tasks > All Tasks**.
2. Click a certification task Number.
3. Enter a new name in the Assigned to field.

Related tasks

- [Activate Data Certification](#)
- [Mark a certification task as closed incomplete](#)

Related concepts

- [Certification schedules](#)
- [Certification tasks](#)
- [Certification elements](#)
- [Certification instances](#)
- [Certification audit instances](#)
- [Certification audit definition](#)
- [Data Certification planning](#)
- [Data certification performance](#)
- [Certification tasks cancellation](#)

- Domain separation and Data Certification

Related reference

- [Data Certification Overview module](#)
- [Send certification task reminders](#)

Send certification task reminders

The Certification Task Escalations workflow sends automatic email reminders.

The Certification Task Escalations workflow sends automatic email reminders to the:

- Certification task owner.
- Assignment group, if the assignment group was specified on the Certification Task form.
- Manager of the certification task owner, if necessary and if a manager was specified on the User form.

The reminders are based on the Complete by field on the certification task record. If the Complete by date is changed, the reminder schedule automatically adjusts to reflect the new date.

Certification task reminders

Time elapsed to end date	Email reminder is sent to	Escalate field on task record reads
50%	task owner and assignment group (if specified)	Moderate
75%	task owner, assignment group, and manager of the task owner	High

Time elapsed to end date	Email reminder is sent to	Escalate field on task record reads
95%	task owner, assignment group, and manager of the task owner	High
100%	task owner, assignment group, and manager of the task owner	High

To set reminders for different or more intervals, edit the workflow Certification Task Escalations. In addition to the email reminders sent automatically, users with the certification_admin role can send email reminders manually at any time.

- [Send an email reminder from the certification task form](#)

How to manually send email reminders from the Certification Task form.

- [Send an email reminder from the certification task list](#)

How to manually send email reminders from the Certification Task list.

Related tasks

- [Activate Data Certification](#)
- [Reassign a certification task](#)
- [Mark a certification task as closed incomplete](#)

Related concepts

- [Certification schedules](#)
- [Certification tasks](#)
- [Certification elements](#)
- [Certification instances](#)

- Certification audit instances
- Certification audit definition
- Data Certification planning
- Data certification performance
- Certification tasks cancellation
- Domain separation and Data Certification

Related reference

- [Data Certification Overview module](#)

Send an email reminder from the certification task form

How to manually send email reminders from the Certification Task form.

Before you begin

Role required: admin

Procedure

1. Navigate to **All > Data Certification > Tasks > All Tasks**.
2. Click a certification task Number.
3. Right-click the header bar and select **Resend email notifications**.

Send an email reminder from the certification task list

How to manually send email reminders from the Certification Task list.

Before you begin

Role required: admin

Procedure

1. Navigate to **All > Data Certification > Tasks > All Tasks**.

2. Select the check box to the left of a certification task Number. Multiple check boxes can be selected.
3. From the Actions on Selected Rows menu below the list, select **Resend email notifications**. Select multiple tasks from the list. The menu option shows how many notifications are outstanding and how many were sent, such as Resend email notifications (15 of 18).

Mark a certification task as closed incomplete

Mark a task as closed incomplete if, for example, only some of the elements can be certified.

Before you begin

Role required: admin

About this task

The following users can mark a task as closed incomplete:

- Users with the certification_admin role.
- User identified in the Assigned to field on the certification task record.

To mark a task as closed incomplete:

Procedure

1. Navigate to **All > Data Certification > Tasks** and select **All Tasks**, or **My Tasks**.
2. Click a certification task Number.
3. In Work Notes, enter information about why the task could not be completed.
4. Click **Close Incomplete**.
If at least one task on a certification instance is marked Closed Incomplete, the Completed date and Percent complete fields on the certification instance record are not updated. A user with the certification_admin role can:
 - Complete the incomplete task or tasks.

- Cancel the incomplete task or tasks.

When all tasks on the certification instance are Closed Complete or Cancelled:

- The system sets the Completed date field on the certification instance record to the current date and time.
- The Percent complete field on the certification instance record is set to 100 percent.

Related tasks

- [Activate Data Certification](#)
- [Reassign a certification task](#)

Related concepts

- [Certification schedules](#)
- [Certification tasks](#)
- [Certification elements](#)
- [Certification instances](#)
- [Certification audit instances](#)
- [Certification audit definition](#)
- [Data Certification planning](#)
- [Data certification performance](#)
- [Certification tasks cancellation](#)
- [Domain separation and Data Certification](#)

Related reference

- [Data Certification Overview module](#)
- [Send certification task reminders](#)

Certification tasks cancellation

Users with the certification_admin role can cancel a certification task in the Work in Progress or Closed Incomplete state.

When a certification task is cancelled, a notification email is sent to the task owner or assignment group assigned to the task. The task owner or assignment group manager is not notified.

- [Cancel an individual task](#)

Cancel a particular data certification tasks in the Work in Progress state.

- [Cancel all tasks in an instance](#)

Cancel data certification tasks in the Work in Progress state.

Related tasks

- [Activate Data Certification](#)
- [Reassign a certification task](#)
- [Mark a certification task as closed incomplete](#)

Related concepts

- [Certification schedules](#)
- [Certification tasks](#)
- [Certification elements](#)
- [Certification instances](#)
- [Certification audit instances](#)
- [Certification audit definition](#)
- [Data Certification planning](#)
- [Data certification performance](#)
- [Domain separation and Data Certification](#)

Related reference

- [Data Certification Overview module](#)
- [Send certification task reminders](#)

Cancel an individual task

Cancel a particular data certification tasks in the Work in Progress state.

Before you begin

Role required: admin

Procedure

1. Navigate to **All > Data Certification > Tasks > All Tasks**.
2. Find a task with a State of Work in Progress.
3. Click the task Number.
4. Click **Cancel**.

Cancel all tasks in an instance

Cancel data certification tasks in the Work in Progress state.

Before you begin

Role required: admin

Procedure

1. Navigate to **All > Data Certification > Schedules > Instances**.
2. Find an instance with a State of Work in Progress.
3. Click the instance Number.
4. Click **Cancel**.

All tasks in the instance with a state of Work in Progress are cancelled. The task owner or assignment group is notified.

The email template used for the notification is named certification.task.cancelled. The email templates can be edited to change the email message text, for example.

Domain separation and Data Certification

Domain separation is supported in Data Certification processing. Domain separation enables you to separate data, processes, and administrative tasks into logical groupings called domains. You can control several aspects of this separation, including which users can see and access data.

Support level: Basic

- Business logic: Ensure data goes into the proper domain for the application's service provider use cases.
- In the application, the user interface, cache keys, reporting, rollups, aggregations, and so on, all consider domain at production run time.
- The owner of the instance needs to be able to set up the application to function normally across multiple tenants.

Use case: When a service provider (SP) uses chat to respond to a tenant-customer's message, the client must be able to see my response.

How domain separation works in Data Certification

- Data Certification has only basic domain separation. As long as the Certification Instances (CIs) or records that must be certified are correctly domain-separated and the users who must certify the CIs or records are in a domain that can view the data, Data Certification works as expected.
- Recommendation: The instance owner must be responsible for assigning Certification Tasks and Certification Instances to the correct domain. Changing the domain for these records does not change functionality, but limits the view of the records.

How to set up domain separation for Data Certification

After enabling the Domain Separation plugin, there are no additional steps required to set up domain separation for Data Certification.

- instance owners determine which CIs or records that need to be certified can be domain-separated.
- Customers can configure a domain-separated environment by assigning tasks to a domain, but if the data is already domain-separated, then only users with the right domain permissions can view the data in a certification task.

How tenant domains manage their own application data

It's not necessary to set the domain on the certification tables but it can be done if the instance owner should want that. As long as the CI's or records that must be certified are domain-separated, users with the correct domain permissions can view them.

Domain-separated tables

- cert_instance – Changing the domain on this table does not change any functionality, nor does it change the domains of the tasks created from the table.
- cert_task – Changing the domain on this table changes the domain viewing permissions of the task.
- cert_element – It is not recommended to change the domain on these records. As long as the CIs or records to be certified are already domain-separated, cert_element records will reflect that.
- cert_filter – Changing the domain on this table changes the domain viewing and filtering of CIs or records.

Use cases

Instance owners who have multiple clients that certify the infrastructure they own can assign domains to those CIs and the Certification Tasks to restrict the view from one client to another.

Related tasks

- [Activate Data Certification](#)
- [Reassign a certification task](#)
- [Mark a certification task as closed incomplete](#)

Related concepts

- [Certification schedules](#)
- [Certification tasks](#)
- [Certification elements](#)
- [Certification instances](#)
- [Certification audit instances](#)
- [Certification audit definition](#)
- [Data Certification planning](#)
- [Data certification performance](#)
- [Certification tasks cancellation](#)

Related reference

- [Data Certification Overview module](#)
- [Send certification task reminders](#)