



HALO ITSM · CMDB &amp; SERVICE MANAGEMENT

# Halo CMDB Configuration Guide

A step-by-step guide to configuring Halo's service-aware CMDB — from enabling the new asset and service linking to building a dependency-driven incident model.

Halo Administrators

CMDB Managers

IT Operations

Implementation Teams

PUBLISHED BY

Allied ESM Ltd

DATE

June 2026

HALO VERSION

Latest stable (hosted)

VERSION

1.0



## FIRST STEPS IN HALO CONFIGURATION

### Two settings unlock the entire CMDB structure.

The service-aware CMDB in Halo is built on a single configuration toggle. Once enabled, Halo connects your CI classes to service catalogue items automatically and activates the four-layer asset hierarchy. These settings are available to all hosted customers on the latest stable release.

Configuration → General Settings → CMDB

1

#### Enable automatic linking of assets and service catalogue

Navigate to **Configuration → General Settings → CMDB**. Tick the option labelled "Allow automatic linking of assets and service catalog". This is still marked as beta in the UI, but is fully stable and available across all hosted instances. Enabling this activates the service-aware asset type settings and makes the hierarchy fields visible across all CI classes.

2

#### Enable environments

On the same settings screen, tick **Enable environments**. Environments allow you to tag Business Application Instances by deployment type — Production, Dev, UAT, Sandbox, and so on. Once enabled, environment values are managed in **Advanced Settings → Lookup Codes**. You can create as many environments as your organisation requires. Enabling environments also allows Business Applications to auto-generate their instances when environments are assigned.

### What these settings change

Once both settings are enabled, the following capabilities become active across the platform.

#### CI CLASSES

##### Service and application flags

Asset type configuration gains new toggles: Is Service, Is Business Application, Is Business Application Instance. These define how each CI class behaves in the hierarchy.

#### SERVICE CATALOGUE

##### Auto-created catalogue items

Creating a Technical Service or Business Application CI now automatically creates a linked service catalogue item — no manual duplication required.

#### RELATIONSHIPS

##### Upstream / downstream hierarchy

Relationship fields (Owning Service, Owning Business Application) become available on each CI class, enabling the full five-layer dependency map to be built and maintained.

#### INCIDENTS

##### Automatic impact calculation

When a CI is linked to a ticket, Halo automatically calculates and populates impacted business applications and services based on the dependency map — no manual field entry needed.

These two settings are a one-time configuration change. Once enabled, all the asset type, relationship, and service catalogue behaviour described in this guide becomes available. They do not affect existing CI data.

## CONFIGURATION → ASSET TYPES

### Four CI classes drive the entire hierarchy.

In Halo, asset types are equivalent to CI classes. The new CMDB structure introduces four specific types — two marked as services and two marked as application tiers. Each is configured via **Configuration → Asset Types**. Halo ships with preconfigured examples of each; you can use these as-is or create new types from scratch using the same settings.

| CI CLASS                     | KEY TOGGLE TO SET              | WHAT THIS CONTROLS                                                                                                                                                       |
|------------------------------|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Business Service</b>      | Is Service = Yes               | Top-level service CI. Creates a hidden service catalogue item automatically. No Owning Service — nothing sits above this layer.                                          |
| <b>Technical Service</b>     | Is Service = Yes               | One level below Business Service. Linked via Owning Service field. Creates a service catalogue item (hidden by default; can be made visible per CI).                     |
| <b>Business Application</b>  | Is Business Application = Yes  | Application CI. Is Service auto-sets to No. Owning Service is required. Environments can auto-generate Business Application Instance CIs from this record.               |
| <b>Business App Instance</b> | Is Business App Instance = Yes | Physical deployment of a Business Application — by environment, region, or business line. Owning Business Application is required. Environment tag available if enabled. |

### Configuring the layout tab for each class

Each asset type has a Layout tab where you control field visibility, whether fields are required, and which fields are not relevant for that class. Getting this right upfront reduces data quality issues later.

#### Business Service and Technical Service

Hide Owning Service and Owning Business Application — nothing sits above a Business Service. Make Business Owner and Technical Owner visible and required. Hide user-allocation fields; these are not user-assigned assets.

#### Business Application

Make Owning Service visible and required. Add a current status field (Fully Online, Degraded, Offline) and any capability or criticality classifications your organisation needs.

#### Business Application Instance

Make Owning Business Application visible and required — this field only returns Business Application CIs. Add the Environment field if environments are enabled. Hide user-association and billing fields.

#### Service type on Technical Services

Add a Service Type field with three values: Application-aligned (e.g. CRM, ERP), Infrastructure-aligned (e.g. Linux Hosting, Network), and End User Delivery (e.g. HQ Service Desk). Make it required.

All standard field customisation is fully available on service and application CI classes. The new hierarchy toggles add to existing asset type configuration — they do not replace it.

## CMDB POPULATION — TOP-DOWN

### Always build the hierarchy from the top down.

Each layer in the CMDB requires its parent to exist before it can be linked. Start with Business Services, then Technical Services, then Business Applications, then Instances. Relationships are set via the Owning Service or Owning Business Application fields on each record — once saved, the link becomes an uneditable upstream relationship on the child CI, which can only be changed by updating the field on the child record.

1

#### Create Business Services

Business Services represent business capabilities — IT Service Management, Finance, Field Sales, Recruiting. Create each as a CI using your Business Service asset type. Assign a Business Owner and Technical Owner. Nothing sits above this layer, so the Owning Service field should be hidden. The Dependencies tab becomes available once downstream Technical Services are linked.

2

#### Create Technical Services and link them up

Technical Services are the automation anchor. Create each one, assign a Service Type (Application-aligned, Infrastructure-aligned, or End User Delivery), and set the Owning Service to the appropriate Business Service. Setting this field creates the upstream relationship automatically. Halo also creates a linked service catalogue item in the background — visible in the service catalogue under the category you specify on the asset type.

3

#### Create Business Applications and link to their Technical Service

For each application (e.g. SAP, Halo, a finance system), create a CI using your Business Application type. Set Owning Service to the Technical Service that manages it. This is required and unambiguous — the field only searches Technical and Business Service CIs. You can also set peer-to-peer relationships between applications that depend on each other (e.g. a finance application that sends data to an ERP via a service bus) using the standard relationships tab.

4

#### Create Business Application Instances

Instances represent the real deployments — Production, Dev, UAT. Set Owning Business Application on each instance record. Every Business Application must have at least one Production instance. To create instances automatically: on the Business Application record, set the environment tags (Production, UAT, Dev) and save — Halo generates the Instance CIs for you. Alternatively, enable the "Automatically create primary instance" toggle on the asset type to have Halo create a Production instance whenever a new Business Application CI is saved.

### Peer relationships and infrastructure

Beyond the parent-child hierarchy, Halo supports same-level relationships between Business Applications and downstream links to Infrastructure CI's. Use the Relationships tab on any CI record to add these. Common relationship types include Sends Data To (application-to-application), Runs On (instance-to-server), and Connected To (server-to-network). These peer and infrastructure relationships appear in the dependency diagram alongside the primary hierarchy and will be included in automatic impact calculations on tickets.

## WHAT THE CMDB ENABLES ONCE BUILT

### The hierarchy does the work — once the data is right.

Once your CI hierarchy is in place, Halo uses it automatically across the service catalogue, incident management, change management, and availability monitoring. The three areas below are the most immediate and high-value outcomes of a correctly configured CMDB.

#### Service catalogue — automatic creation and linking

##### Technical Services create hidden catalogue items

When you create a Technical Service CI, Halo automatically creates a linked service catalogue item in the background. This item is hidden from end users by default — it exists for agent use and internal reporting. You can change visibility per CI if you want end users to raise requests directly against a Technical Service.

##### Business Applications create visible catalogue items

Business Application CIs create a visible catalogue entry by default — the category it appears under is set on the asset type configuration. This gives you a twin: the CI for CMDB/dependency tracking, and the catalogue item for service requests and user-facing forms. Change the visibility or add request forms via the Service View on the CI record.

#### Incidents — automatic impact calculation

When a CI is set as the primary configuration item on an incident (at creation, during triage, or at any point in the ticket lifecycle), Halo reads the dependency map for that CI and automatically populates two fields on the incident form: **Impacted Business Applications** and **Impacted Services**. Changing the primary CI recalculates these fields instantly. You can also set the health status on a CI directly from the incident record to trigger status notifications on the self-service portal.

##### Using impacted fields for automation

The Impacted Business Applications and Impacted Services fields can drive notification rules — automatically notifying the Business Owner of any application in the impacted list. To enable this: add the impacted fields to your incident form layout, then configure notification delivery methods to reference them as recipients.

#### Service availability monitoring

Available on both Technical Services and Business Applications. Set a target uptime percentage on the CI record, then configure rules to define which tickets count as downtime — based on ticket type, priority, or any other attribute. When a matching ticket closes, Halo calculates the elapsed time and updates the availability score automatically. A manual **Is Downtime** field can also be added to the ticket form for case-by-case flagging outside of automated rules.

The asset import spreadsheet supports all hierarchy fields — Owning Service, Owning Business Application, environments — for bulk-loading CMDB data. The same fields are available via API integrations.

## PURE-PLAY HALO PARTNER

## Allied ESM is 100% focused on Halo.

We are one of a small number of pure-play Halo ITSM partners globally. Every consultant, every project, every support call is built exclusively around Halo — no competing platforms, no divided loyalties.

CMDB configuration is one of the areas where we see the most variance between Halo implementations. Getting the CI hierarchy, ownership model, and service catalogue integration right from the start saves significant rework down the line. We help teams do it properly.

## WHAT ALLIED ESM DELIVERS

- Implementation — fixed-price plans from 5 weeks (Micro) to 17 weeks (Complex)
- CMDB architecture — CI class design, hierarchy setup, relationship mapping
- SAF configuration — Service Data Model and Target Operating Model where scope allows
- Change and adoption — embedding CMDB governance and data ownership across your org
- Managed services — ongoing expert support with a named dedicated team
- ITSM staffing — UK-based Halo-certified consultants for your team

## WEBSITE

[alliedesm.com](https://alliedesm.com)

## GET IN TOUCH

[info@alliedesm.com](mailto:info@alliedesm.com)

## HALO IMPLEMENTATION

[alliedesm.com/services](https://alliedesm.com/services)

## CMDB &amp; SAF

[alliedesm.com/halo](https://alliedesm.com/halo)