



# VOSS Automate Best Practices Guide

Release 25.1

June 05, 2025

## Legal Information

- Copyright © 2025 VisionOSS Limited.  
All rights reserved.
- This information is confidential. If received in error, it must be returned to VisionOSS ("VOSS"). Copyright in all documents originated by VOSS rests in VOSS. No portion may be reproduced by any process without prior written permission. VOSS does not guarantee that this document is technically correct or complete. VOSS accepts no liability for any loss (however caused) sustained as a result of any error or omission in the document.

DOCUMENT ID: 20250605105622

# Contents

<b>1</b>	<b>What's New</b>	<b>1</b>
1.1	Best Practices Guide: Release 25.1 . . . . .	1
<b>2</b>	<b>Use the Action search to navigate Automate</b>	<b>2</b>
2.1	Overview . . . . .	2
2.2	Best practices for meaningful Action search results . . . . .	4
<b>3</b>	<b>Deployment</b>	<b>5</b>
3.1	Automate deployment topologies . . . . .	5
<b>4</b>	<b>Deployment models and web weight settings</b>	<b>23</b>
4.1	Overview . . . . .	23
4.2	Active-Active web weights . . . . .	23
4.3	Active-StandBy web weights . . . . .	24
<b>5</b>	<b>Overload controls</b>	<b>25</b>
5.1	Session limits . . . . .	25
5.2	Throttle limits . . . . .	25
5.3	Configurable number of queue processes . . . . .	26
<b>6</b>	<b>Onboarding customers and users</b>	<b>28</b>
6.1	Planning guidelines for onboarding and ongoing operations . . . . .	28
<b>7</b>	<b>Webex onboarding best practices</b>	<b>29</b>
7.1	Add Automate over an existing Webex organization with configuration already in place . . . . .	29
7.2	Add Automate management over an existing, un-provisioned Webex organization . . . . .	33
7.3	Webex syncs . . . . .	39
<b>8</b>	<b>Data sync</b>	<b>40</b>
8.1	General sync principles and best practices . . . . .	40
8.2	Cisco UCM . . . . .	45
8.3	Cisco Unity Connection . . . . .	49
8.4	LDAP . . . . .	50
8.5	Cisco Webex App (Spark) . . . . .	50
8.6	Microsoft Graph and Teams sync . . . . .	51
<b>9</b>	<b>Data collection</b>	<b>69</b>
9.1	Recommended RIS API data collector interval . . . . .	69
<b>10</b>	<b>API performance</b>	<b>70</b>
10.1	API resource listing best practice . . . . .	70
10.2	Long running API requests . . . . .	71

<b>11 System maintenance</b>	<b>73</b>
11.1 Transaction archiving . . . . .	73
11.2 Automated database cache cleanup . . . . .	73
<b>12 Admin Portal setup</b>	<b>75</b>
12.1 Navigation - menu and dashboards . . . . .	75
<b>Index</b>	<b>79</b>

# 1. What's New

## 1.1. Best Practices Guide: Release 25.1

- EKB-21976: Create replacement feature for syncing CsOnlineUsers based on synced MsolUsers. See: [Microsoft syncs](#)  
Updated Microsoft sync docs for replacement auto-filter Teams/CsOnlineUser sync
- EKB-21976: Create replacement feature for syncing CsOnlineUsers based on synced MsolUsers. See: [Configure Microsoft tenant connection parameters](#)  
Updated Microsoft sync docs for replacement auto-filter Teams/CsOnlineUser sync

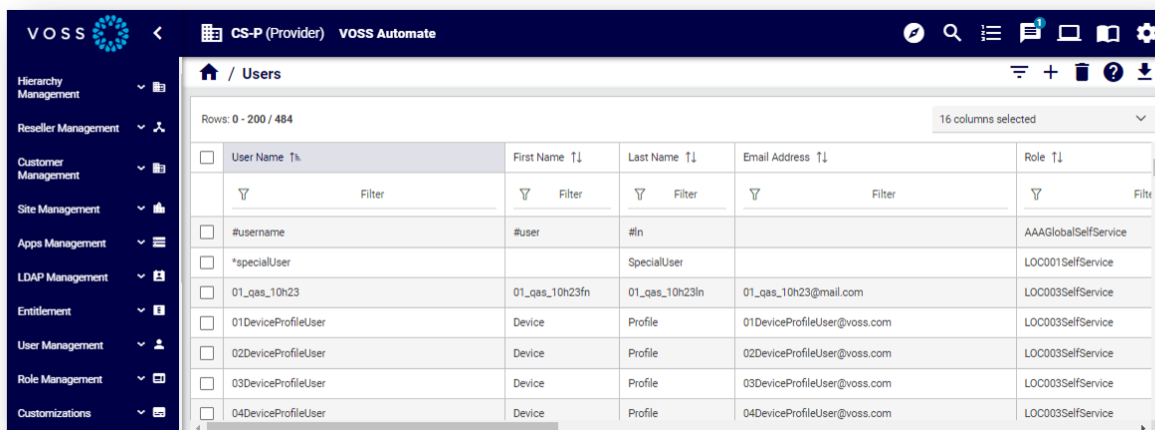
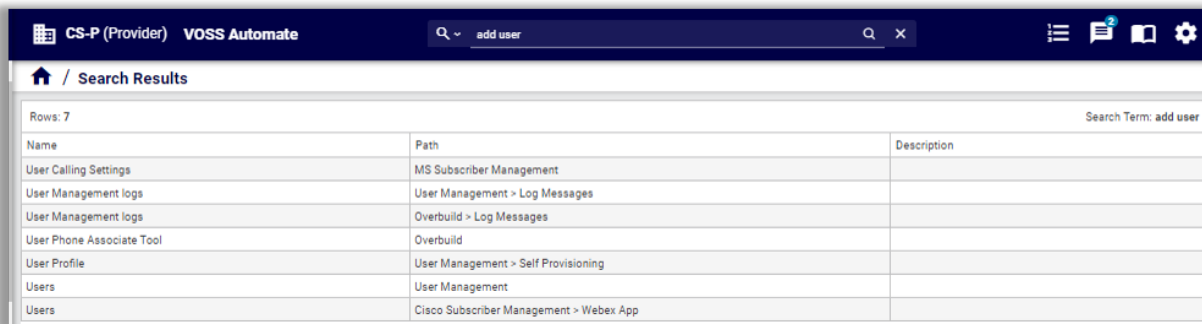
## 2. Use the Action search to navigate Automate

### 2.1. Overview

Automate's toolbar Action search allows you to navigate the GUI to go to pages in the user interface.

**Note:** Automate 25.1 replaced the default menu layout with a series of dashboards and menus to consolidate Automate functionality based on a role-based way of working for different administrators, including Provider admin, Customer admin, Site admin.

The Action search is the default search mode in Automate. Fill out the name of the page you're looking for, then press **Enter** or click the Search icon. Click on the relevant search result to open the page.



- Menu entry results display in the following **Path** format: *xx > xx*
- Dashboard results display in the following **Path** format: *Dashboard: yyy*
- Verbs in the search phrase return results when matching the same word in a menu or dashboard (for your role).
- Since the Action search uses title text in menus, Dashboards, and actions, a best practice when creating these is to ensure they contain useful text for finding actions.

---

**Important:**

- Plural keywords may not return all relevant results. If you don't see the results you expect, change the keyword to singular, for example, *phone* instead of *phones*, or *add subscriber*, instead of *add subscribers*.
  - Using generic terms, such as *subscriber* or *phone*, returns all items relating to this keyword.
  - Avoid search phrases that include *the*, *a* or *an*. For example, use *add user* instead of *add a user*.
  - Search phrases that refer to a device, such as Cisco UCM (Cisco Unified Communications Manager), returns all items (including device models) that have the device name in the label or description. Additionally, the phrase *add device* returns a list of results where the label starts with *device*, where the role allows the add operation, and (in this case), labels, descriptions, or models containing the full string, *add device*.
  - For best results (depending on the permissions you have on the model types), use the following verbs in search phrases:
    - create, add
    - update, edit, modify, change
    - delete, remove
- 

Search criteria supports abbreviations, model type keywords (such as view, relation, model), and matches for the first character of words in a string. For example, *QAS* returns *QAS - MS Teams*, as well as *Quick Add SIP Gateway*.

**Note:** Including numerical digits in the search criteria only returns menus with digits in the menu or dashboard name. For example, search criteria including the digits *1*, *6*, or *4* returns menus with these digits in their name, such as *E164 Inventory* (by default, a sub-menu in the **Number Management** menu).



The screenshot shows the VOSS Automate interface with a search bar containing '64'. Below the search bar, there is a table titled 'Search Results' with 4 rows. The table has three columns: Name, Path, and Description. The search term '64' is displayed in the top right corner of the table area.

Name	Path	Description
Add E164 Inventory	Number Management	
E164 Associations ( N to 1 DN )	Number Management	
E164 Associations ( N to N DN )	Number Management	
E164 Inventory	Number Management	

## 2.2. Best practices for meaningful Action search results

Administrators should align naming conventions and the setup of items for user navigation with terms that are familiar to users. For example:

- Adjust the names of menus, dashboards, tasks, and quick actions if these prove to be unintuitive.
- Configure relevant dashboards for quick actions, based on user roles.
- Use the Action search to quickly find actions that aren't available on dashboards and menus.

### Related topics

- Search in VOSS Automate in the Core Feature Guide



## 3. Deployment

### 3.1. Automate deployment topologies

#### 3.1.1. Overview

Automate offers two main deployment topologies:

- *Unified node cluster topology*
- *Modular node cluster deployment topology*

Two additional deployment options are available:

- *Cloud deployments*
- *VOSS Automate Cloudv1 (SaaS)*

#### 3.1.2. Node types

Automate deployment topologies are comprised of a configuration of the following types of nodes, each performing specific functions within the topologies:

- Web proxy node
- Unified/single node
- Application node
- Database node

Each node type is comprised of one or more of the following components (software subsystems):

Component	Description
Operating system	Ubuntu, stripped down / hardened
Platform	Docker, isolated components
Web server	Nginx, receives and forwards HTTP requests <ul style="list-style-type: none"> <li>• Hosts static files: CSS, JS and images</li> <li>• Load balance between unified nodes (UNs): round robin, configurable, for example, two data centres</li> <li>• Detects inactive UN: removes from round robin</li> </ul>
Database	MongoDB (scalable, distributed), PostgreSQL (scalable)
Application	JavaScript, Python, REST API, device drivers, workflow engine, transactions/queue engine, RBAC, search, bulk loader, and more ...

The matrix outlined in the table describes the set of components in each node type:

Node type	Components				
	Operating system	Platform	Web server	Database	Application
Web proxy	X	X	X		
Unified/single node	X	X	X	X	X
Application	X	X			X
Database	X	X		X	

### 3.1.3. Unified node cluster topology

Automate's **Unified Node Cluster** topology provides the following options:

- *Single-node cluster (cluster-of-one/standalone) (testing-only)*
- *Single-node cluster (cluster-of-one/standalone) with VMWare HA*
- Two node with web proxies
- Four node with web proxies
- Six node with web proxies

---

**Important:** Choose between a Unified Node deployment or a Modular Architecture deployment.

---

In a *Unified Node Cluster* deployment, Automate is deployed as *one* of the following:

- A single unified node cluster
- Two unified nodes
- A cluster of multiple nodes with High Availability (HA) and Disaster Recovery (DR) qualities

Each node can be assigned one or more of the following functional roles:

Functional role	Description
Web proxy	Load balances incoming HTTP requests across unified nodes.
Single unified node	Combines the Application and Database roles for use in a non-multi-clustered test environment.
Unified	Similar to the <i>Single unified node</i> role Application and Database roles, but clustered with other nodes to provide HA and DR capabilities.

The nginx web server is installed on the web proxy, *Single Unified Node*, and the *Unified Node Cluster*, but is configured differently for each role.

In a clustered environment containing multiple *Unified Node Clusters*, a load balancing function is required to offer HA (High Availability providing failover between redundant roles).

Automate supports deployment of either the web proxy node or a DNS load balancer. Consider the following when deciding whether to select a web proxy node or a DNS:

- The web proxy node takes load off the *Unified Node Cluster* to deliver static content (HTML/JAVA scripts). When using DNS or a third-party load balancer, the *Unified Node Cluster* must process this information.
- DNS is unaware of the state of the *Unified Node Cluster*.
- The web proxy detects if a *Unified Node Cluster* is down or corrupt. In this case, the web proxy will select the next *Unified Node Cluster* in a round robin scheme.

---

**Important:** It is recommended that you run no more than two *Unified Node Clusters* and one web proxy node on a physical (VMware) server.

Additionally, it is recommended that the disk sub-systems are unique for each *Unified Node Cluster*.

---

The table describes the defined deployment topologies for test and production:

Deployment topology	Description
Test	<p>A standalone, <i>Single Unified Node</i>, with Application and Database roles combined.</p> <p>No high availability or disaster recovery (HA/DR) is available.</p> <hr/> <p><b>Important:</b> A test deployment must be used only for test purposes.</p> <hr/>
Production with unified nodes	<p>In a clustered system, comprising:</p> <ul style="list-style-type: none"><li>• Two, three, four, or six unified nodes (each with combined Application and Database roles)</li><li>• Zero to four (maximum two if two unified nodes) web proxy nodes offering load balancing.</li></ul> <p>The web proxy nodes can be omitted if an external load balancer is available.</p>

#### Single-node cluster (cluster-of-one/standalone) (testing-only)

**Note:** A *Single-node cluster (cluster-of-one/standalone)* deployment should be used *only* for test purposes.



The table describes the advantages and disadvantages of a *Single-node cluster (cluster-of-one/standalone)* deployment topology:

Advantages	Disadvantages
<ul style="list-style-type: none"> <li>• Smallest hardware footprint</li> </ul>	<ul style="list-style-type: none"> <li>• No high availability or disaster recovery</li> <li>• Less throughput than clusters</li> </ul>

### Single-node cluster (cluster-of-one/standalone) with VMWare HA

The table describes the advantages and disadvantages of a *Single-node cluster (cluster-of-one/standalone)* with VMWare HA deployment topology:

Advantages	Disadvantages
<ul style="list-style-type: none"> <li>• Smallest hardware footprint</li> <li>• Disaster recovery available</li> </ul>	<ul style="list-style-type: none"> <li>• Less throughput than clusters</li> </ul>

### Multi-node cluster with unified nodes

To achieve geo-redundancy using the unified nodes, consider the following:

- Either four or six unified nodes (each node combining Application and Database roles), are clustered and split over two geographically disparate locations.
- Two web proxy nodes to provide high availability, ensuring that an Application role failure is gracefully handled. More may be added if web proxy nodes are required in a DMZ.

---

**Important:** It is strongly recommended *not* to allow customer end-users the same level of administrator access as the restricted groups of Provider- and Customer administrators. For this reason, Self-service web proxies as well as Administrator web proxies should be used.

Systems with Self-service-only web proxies are *only* recommended where the system is customer facing, but where the customer does not administer the system themselves.

---

- Web proxy and unified nodes can be contained in separate, firewalled networks.
- Database synchronization takes places between all database roles, thus offering disaster recovery and high availability.
- For six unified nodes, all nodes in the cluster are active. For an eight node cluster (with latency between data centers greater than 10ms), the two nodes in the disaster recovery node are passive; that is, the `voss workers @ command` has been run on the disaster recovery nodes.

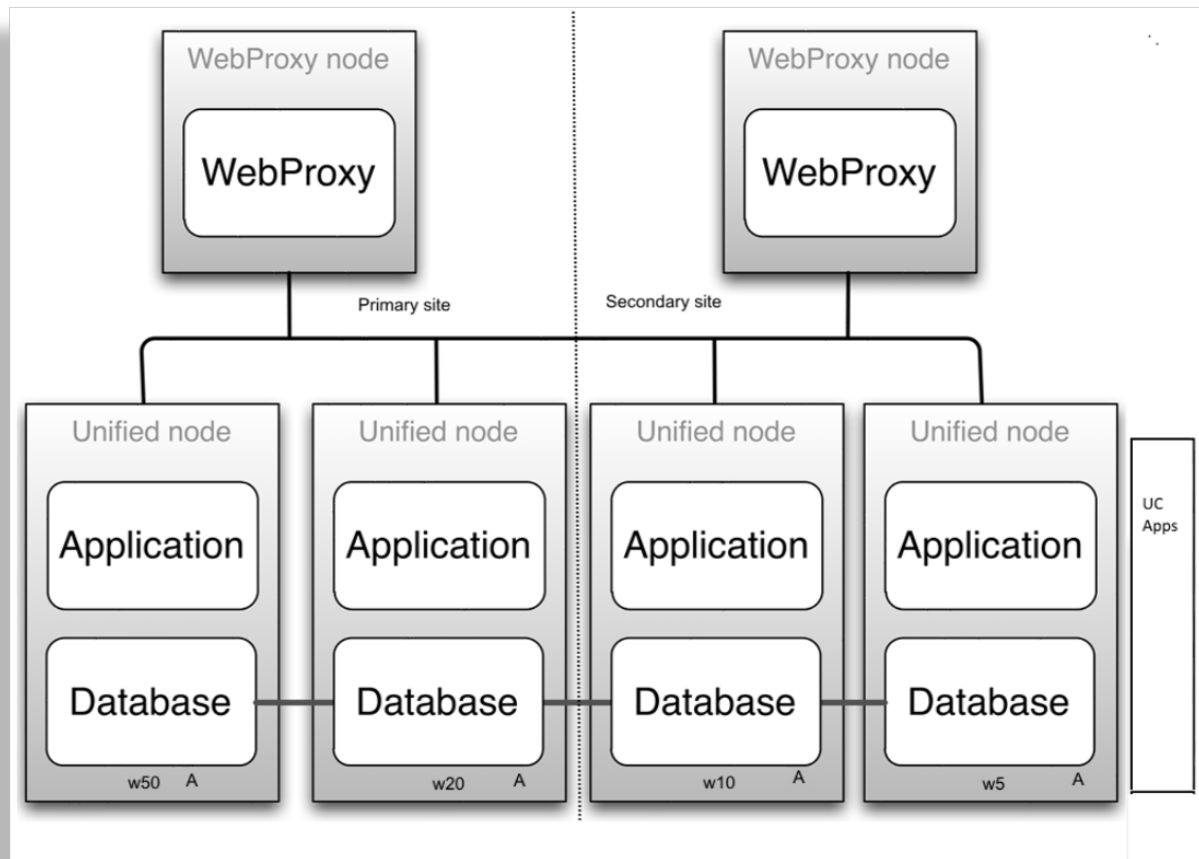
---

**Note:** Primary and fall-back secondary database servers can be configured manually. Refer to the *Automate Platform Guide* for details.

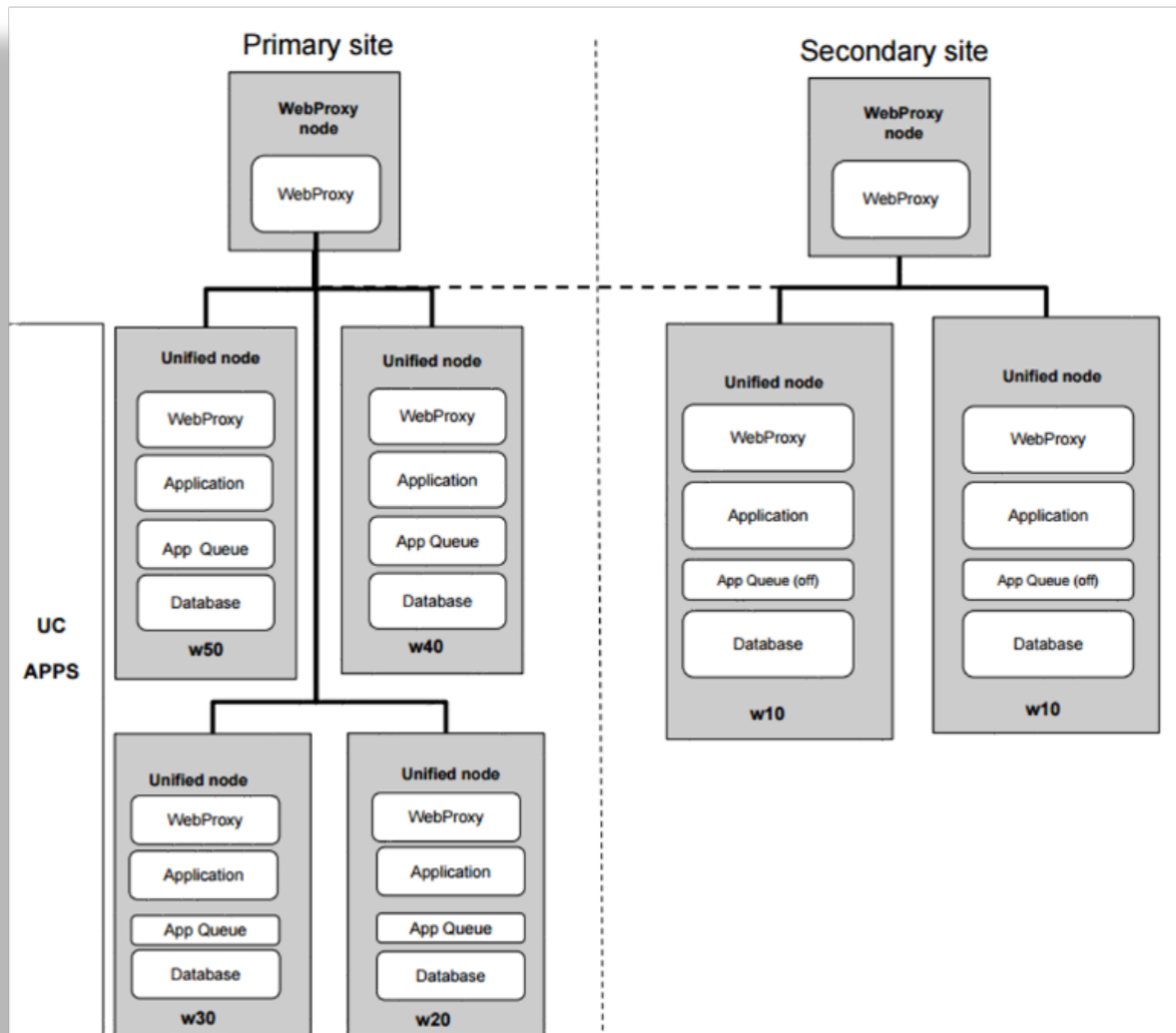
---

**Example: Six node cluster**

The diagram illustrates an example of a *six node cluster*:

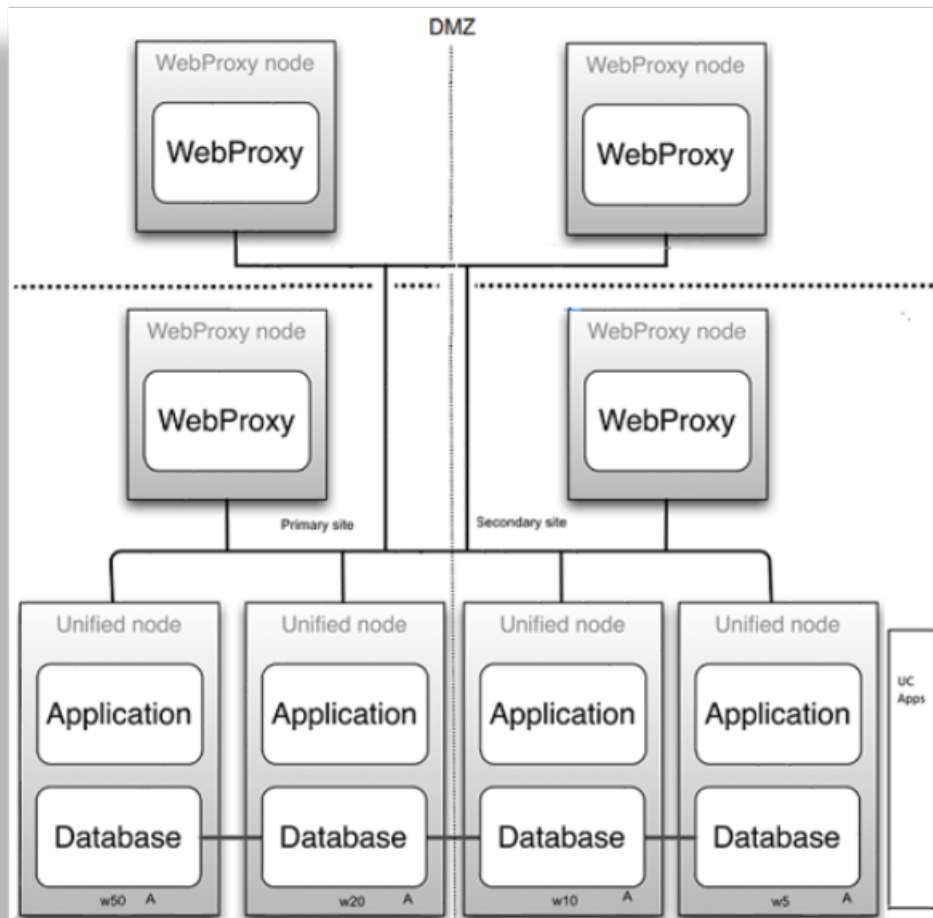
**Example: Eight node cluster**

The diagram illustrates an example of an *eight node cluster*:



#### Example: Two web proxy nodes in a DMZ

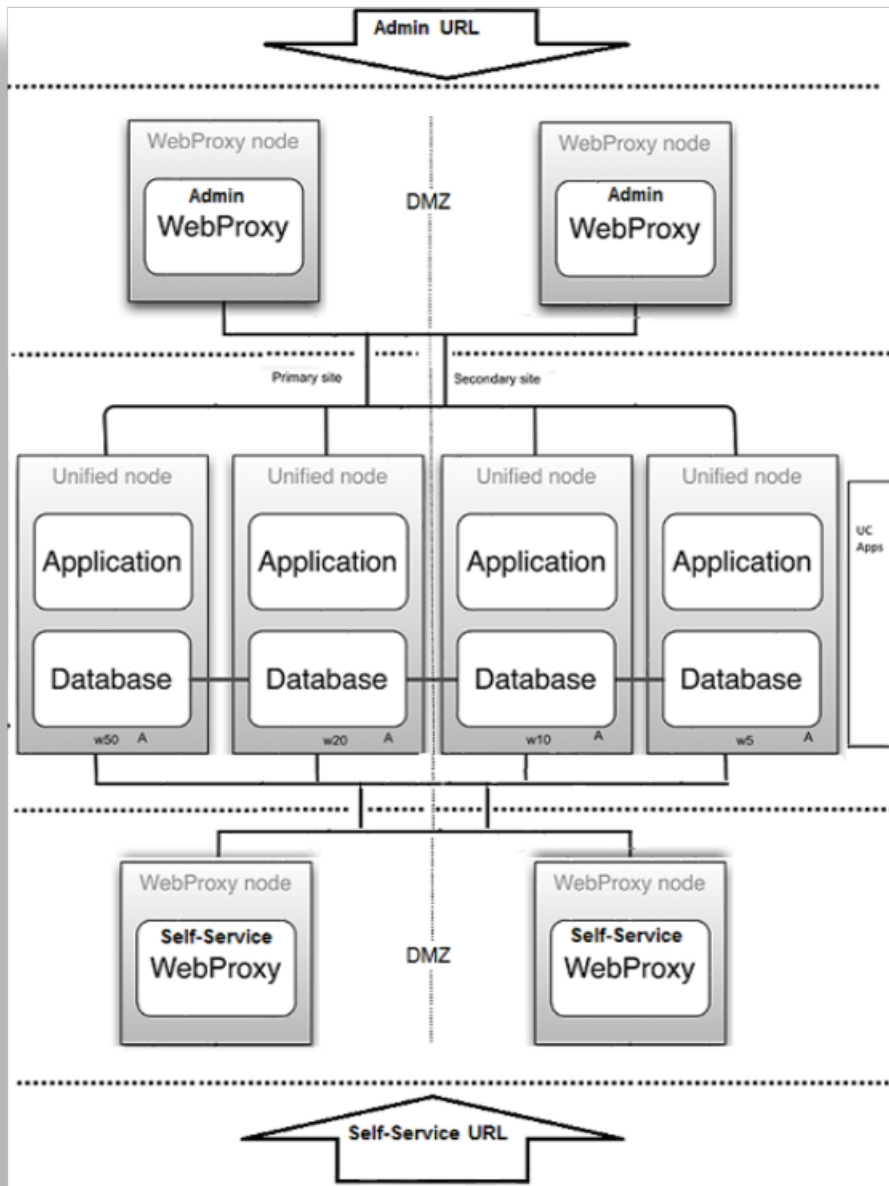
The diagram illustrates an example of *two web proxy nodes in a DMZ*:



**Example: Four web proxy nodes in a DMZ (two admin, two Self-service)**

The diagram illustrates an example of *four web proxy nodes (2 admin, and 2 Self-service)* in a DMZ:





### Two node cluster with unified nodes

To achieve geo-redundancy using the unified nodes, consider the following:

- Two unified nodes (each node combining application and database roles) are clustered and optionally split over two geographically disparate locations.
- (Optional) Two web proxy nodes can be used. It may be omitted if an external load balancer is available.
- Web proxy and unified nodes can be contained in separate firewalled networks.
- Database synchronization takes place from primary to secondary unified nodes, thereby offering disaster recovery if the primary node fails.

- If the secondary unified node has *more than 10ms latency* with the primary unified node, it must be configured to be in the *same* geographical location.

---

**Important:**

With only two unified nodes, with or without web proxies, there is no high availability. The database on the primary node is read/write, while the database on the secondary is read-only.

Only redundancy is available in the following instances:

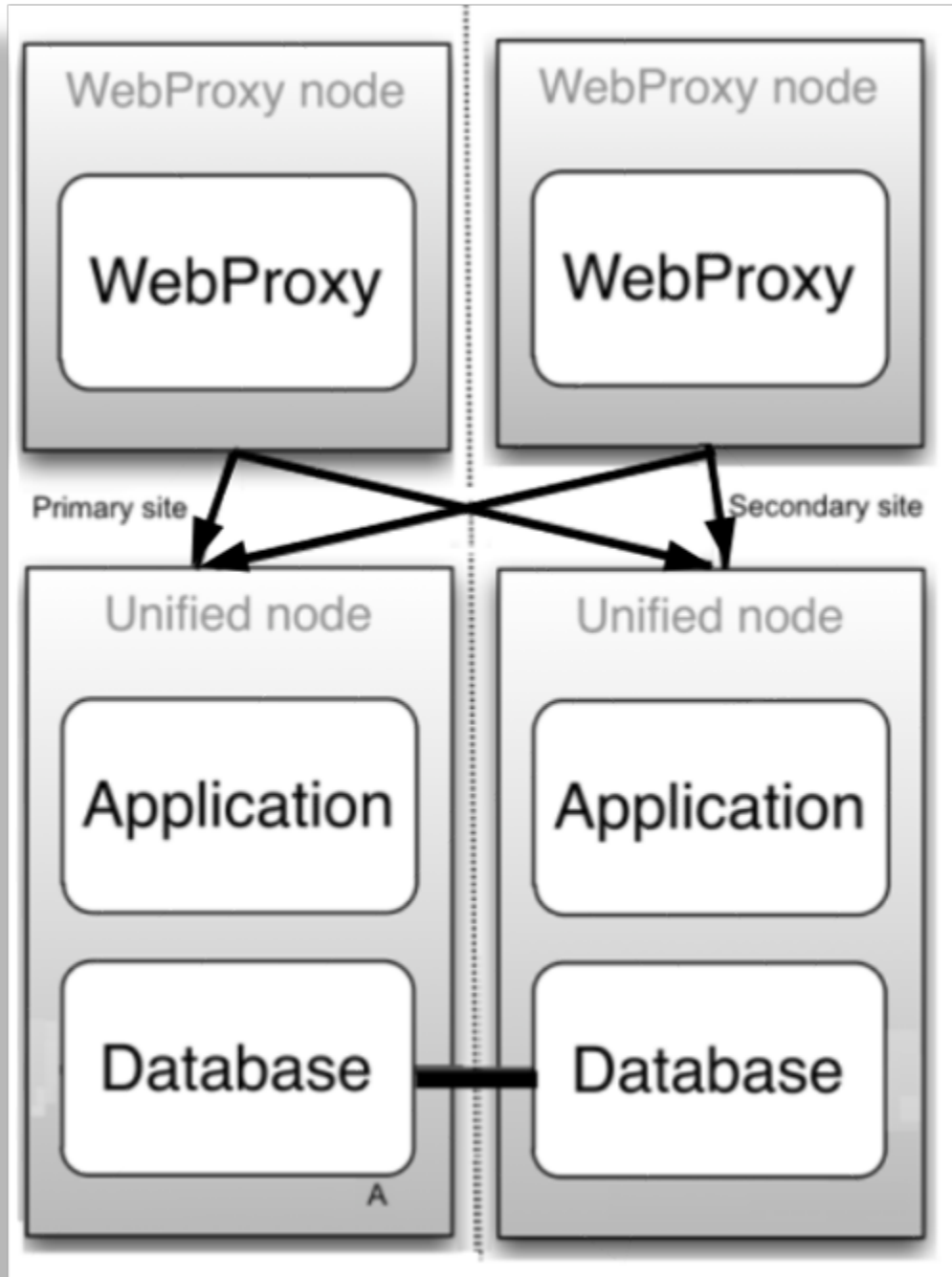
- If the primary node fails, a manual delete of the primary node on the secondary and a cluster provision will be needed.
- If the secondary node fails, it needs to be replaced.

Refer to the topic on *Disaster recovery failover and recovery in a two node cluster* in the Platform Guide.

---

**Example: Two node cluster**

The diagram illustrates a *two node cluster*:



### Four node with web proxies

The table describes the advantages and disadvantages of a *four node with web proxies* deployment topology:

Advantages	Disadvantages
<ul style="list-style-type: none"> <li>• More disaster recovery scenarios supported</li> <li>• More throughput than 3 Node</li> </ul>	<ul style="list-style-type: none"> <li>• More hardware than 3 Node</li> </ul>

### Six node with web proxies

The following are characteristics of a *six node with web proxies* deployment topology:

- Typically deployed for multi-data center deployments
- Supports Active/Standby

## 3.1.4. Modular node cluster deployment topology

### Overview

A *modular node cluster* topology has separate Application and Database nodes:

- Three Database nodes
- One to eight Application nodes
- Web proxies

A *modular node cluster\** topology has the following advantages:

- Increased processing capacity
- Horizontal scaling by adding more Application nodes
- Improved database resilience with dedicated nodes and isolation from application
- Improved database performance by removing application load from the primary database

---

**Important:** Choose between a *Unified Node Cluster* deployment or a *Modular Node Cluster* deployment.

---

Automate is deployed as a *Modular Node Cluster* of multiple nodes, with High Availability (HA) and Disaster Recovery (DR) qualities.

Each node can be assigned one or more of the following functional roles:

Functional role	Description
Web proxy	Load balances incoming HTTP requests across nodes.
Application role node	Clustered with other nodes to provide HA and DR capabilities.
Database role node	Clustered with other nodes to provide HA and DR capabilities.

The nginx web server is installed on the web proxy and application role node, but is configured differently for each role.

### Related topics

Modular Architecture Multi-node Installation in the Install Guide.

Migrate a Unified Node Cluster to a Modular Node Cluster in the Platform Guide.

A load balancing function is required to offer HA (High Availability providing failover between redundant roles).

Automate supports deployment of either the web proxy node or a DNS load balancer. When choosing between a web proxy node and a DNS, consider the following:

- The web proxy takes load off the Application role node to deliver static content (HTML/JAVA scripts). When using DNS or a third-party load balancer, the Application role node has to process this information.
- DNS is unaware of the state of the Application role node.
- The web proxy detects if an Application role node is down or corrupt. In this case, the web proxy will select the next Application role node in a round robin scheme.

**Important:** It is recommended that you run no more than one Application role node and one Database role node and one web proxy node on a physical (VMWare) server. When choosing disk infrastructure, high volume data access by database role replica sets must be considered where different disk sub-systems may be required depending on the performance of the disk infrastructure.

The following *Modular Node Cluster* topology is recommended (minimum):

**Important:** *Single Unified Node* topologies are not available for *Modular Node Cluster* deployments.

- Production with nodes (in a clustered system of two data centers):
  - DC1 = Data center 1, a primary data center containing primary database node (highest database weight)
  - DC2 = Data center 2, a data recovery data center

The system comprises of the following nodes:

- Three nodes with application roles (two in DC1; one in DC2)
- Three nodes with database roles (two in DC1; one in DC2)

- Maximum two web proxy nodes if two data centers; offering load balancing. The web proxy nodes can be omitted if an external load balancer is available.

### Multi-node modular node cluster with application and database nodes

To achieve geo-redundancy using Application and Database nodes, consider the following:

- Six Application and Database nodes (three nodes with an application role and three nodes with a database role) are clustered and split over two geographically disparate locations.
- Two web proxy nodes to provide High Availability so that an Application role failure is gracefully handled. More may be added if web proxy nodes are required in a DMZ.

---

**Important:** It is strongly recommended *not* to allow customer end-users the same level of administrator access as the restricted groups of Provider- and Customer administrators. For this reason, Self-service web proxies as well as Administrator web proxies should be used.

Systems with Self-service-only web proxies are *only* recommended where the system is customer facing, but where the customer does not administer the system themselves.

---

- Web proxy, Application and Database nodes can be contained in separate firewalled networks.
- Database synchronization takes places between all database role nodes, thus offering disaster recovery and high availability.
- All nodes in the cluster are active.

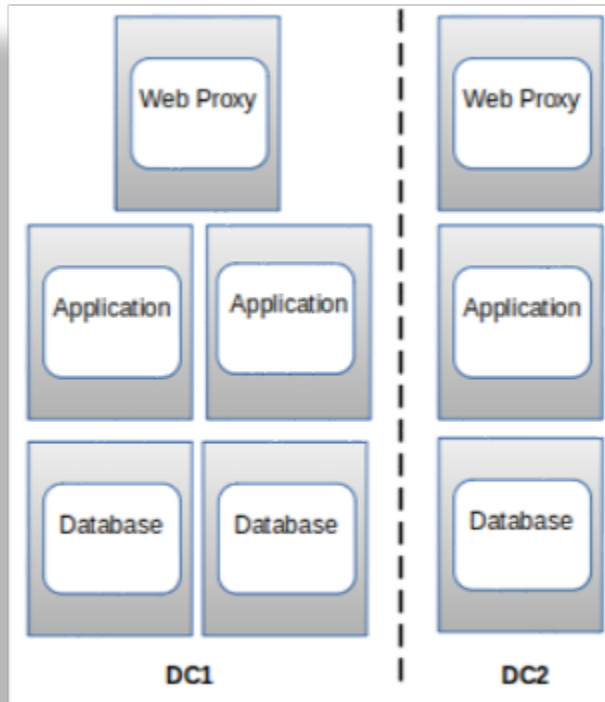
---

**Note:** Primary and fall-back secondary database servers can be configured manually. Refer to the *Automate Platform Guide* for details.

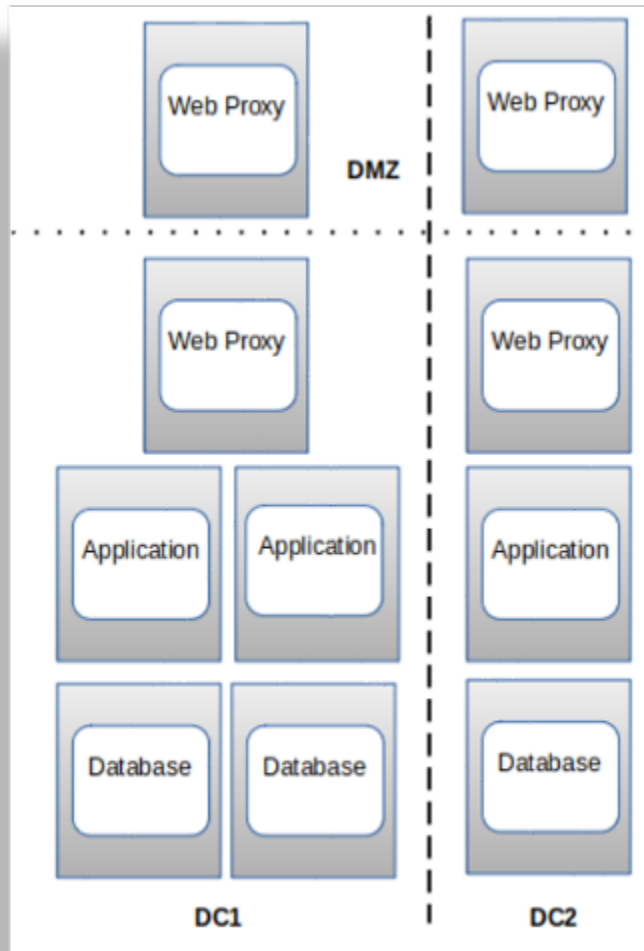
---

### Example: Six node cluster

The diagram illustrates an example of a *six node cluster*:

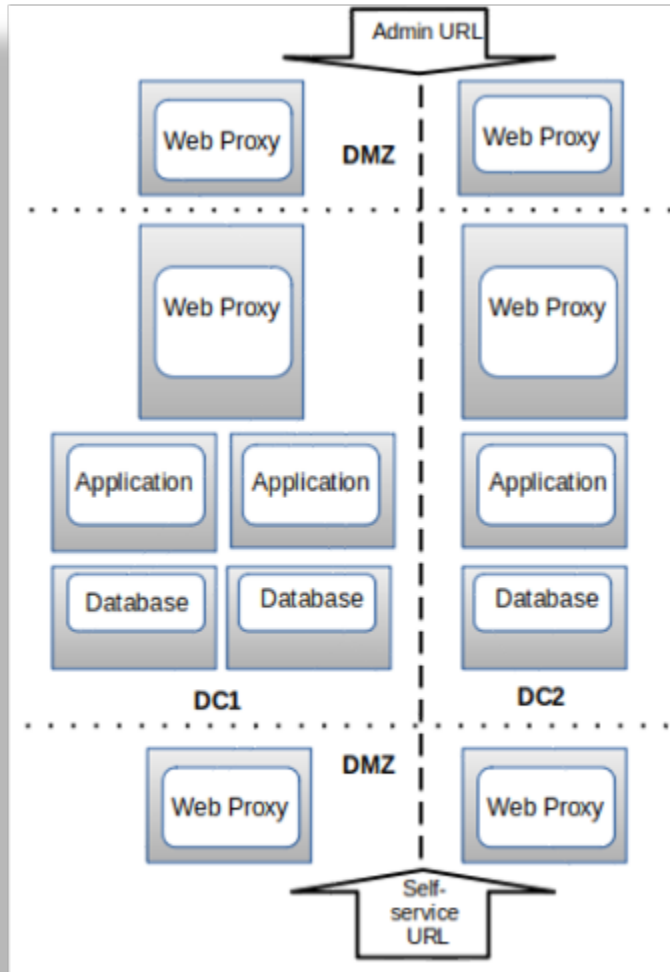
**Example: Two web proxy nodes in a DMZ**

The diagram illustrates an example of *two web proxy nodes in a DMZ*:

**Example: Four web proxy nodes in a DMZ**

The diagram illustrates an example of *four web proxy nodes in a DMZ* (two admin, two Self-service):





### 3.1.5. Cloud deployments

Automate supports the following Cloud deployments:

- Microsoft Azure
- Amazon Web Services (AWS)

Although Google Cloud Platform (GCP) is not officially supported, contact us to discuss your requirements.

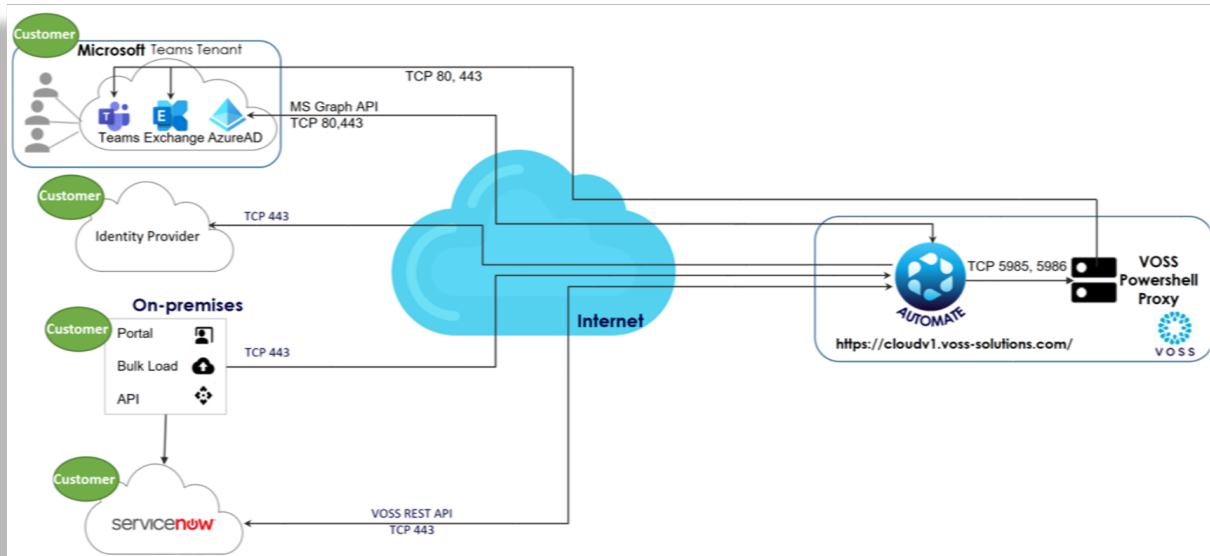
Advantages of a Cloud deployment topology:

- Leverage cloud tooling, such as proxies (which can be used instead of VOSS Web Proxy)

### 3.1.6. VOSS Automate Cloudv1 (SaaS)

VOSS Automate Cloudv1 is a Software-as-a-Service (SaaS) offering hosted on a shared VOSS Automate instance within Microsoft Azure.

VOSS manages this instance, which seamlessly integrates with a customer's Unified Communications (UC) platform, Microsoft Exchange, Microsoft Active Directory, and third-party applications, such as ServiceNow and Identity Providers (IdPs) for Single Sign-On (SSO) authentication.



## 4. Deployment models and web weight settings

### 4.1. Overview

- The supported deployment models are described in the Install Guide under Chapter 2 Deployment Topologies.
- Web weights are explained in the Install Guide under Multi Data Center Deployments and Multinode Installation. The web weight specifies the routing and relative counts of the initial HTTP request from the Web Proxy to a Unified Node. The initial request could be a request such as a transaction, or for example a GET request.
- Transactions can be processed on *any* Active Unified Node - regardless of which Unified Node processed the initial HTTP request. The transaction log provides the detailed information in the fields shown below:
  - `submitter_host_name`: the hostname of the application node that scheduled the transaction.
  - `processor_host_name`: the hostname of the application node that processed the transaction (this value is set once the transaction is processed).
- Note that a sub-transaction can be processed on a different Unified Node than its immediate precedent in the hierarchy.
- We recommend that you use both Web Proxies. However, the use of only 1 Web Proxy is supported (and Web Proxy use is optional).
- To display the configured web weights, run the command **web weight list** at the CLI of each Web Proxy Node .
- The recommended web weight settings for the various deployment models are shown in the following sections.

### 4.2. Active-Active web weights

- There are four active unified nodes, two in each data center. The maximum supported Round Trip Time (RTT) is 10ms.
- WP1: 1 1 0 0
- WP2: 0 0 1 1

This scheme is designed to route the initial HTTP request to the unified nodes local to the Web Proxy Node that forwards the request. If only one Web Proxy (WP) is used, then use the following setting for WP1:

WP1: 1 1 1 1

This results in some of the initial HTTP requests crossing to the secondary data center, however this has a minimal impact on system performance.

### 4.3. Active-StandBy web weights

- There are four active unified nodes in the primary data center and two standby unified nodes in the secondary data center. The maximum supported RTT is 400ms.
- WP1: 1 1 1 1 0 0
- WP2: 1 1 1 1 0 0

If only Web Proxy one (WP1) is used, the default weights provided by the system are sufficient. If Web Proxy two (WP2) or both Web Proxy Nodes are used, change the web weights at Web Proxy two (WP2) to the values noted above.

The logic behind the web weight settings for the Active-StandBy model is that some non-transaction work may generate a significant number of queries back to the Primary DB. Therefore, processing such work in the secondary data center may result in unacceptably long processing times.

## 5. Overload controls

### 5.1. Session limits

The following numbers represent the default and maximum values:

- global administration: 200 (includes non-customer admins as well as service provider and reseller admins). This limit also includes API clients configured as Admin Users.
- global self-service: 20,000 - This is the total number of self-service users logged into the system - both active and inactive.
- per customer administration: 10 (this should be set to a lower value in some cases). The Partner must first “reserve” a number of non-customer admin sessions from the global limit of 200 noted above. The remaining admins can be allocated to customers. Based on the expected number of customers, the Partner can then set the per customer admin limit.

For example, if the Partner wishes to “reserve” 20 admin sessions for non-customer use, that would leave 180 available for customer use. If a total of 40 customers is expected, the Partner should set the per customer admin limit to no more than  $180/40 = 4$  (rounded down from 4.5). In this example, a maximum of  $40 \times 4 = 160$  admin sessions can be allocated to customer-level admins. This would effectively reserve  $200 - 160 = 40$  admins for the Partner to use.

- per customer self-service: 1000

### 5.2. Throttle limits

- Admin (by default, this is disabled). It is recommended that the Admin throttle is enabled and set to 450 API requests/min. The setting is per unified node.
  - Service Inventory (SI) Report: Relies on the per-user throttle setting to ensure adequate throughput.
    - \* For the Active-StandBy deployment model, it is strongly recommended that the SI report is configured to run on a specific non-primary unified node (preferably in the Secondary Data Center as those nodes are likely to have a lower load). This results in faster performance, but there is not any protection against a single node failure in the middle of an SI report run (not very likely). The use of a Web Proxy is *not* recommended here as 25% of the initial requests are routed to the Primary Unified Node based on the recommended web weight settings at either Web Proxy.
    - \* For the Active-Active deployment model, you can use a Web Proxy instead. The SI report would run slower, but this configuration provides protection against a single Unified Node failure during the SI report run. If a Web Proxy is used, then:

- Use Web Proxy 2. This prevents routing to the Primary Unified Node based on the recommended web weight settings.
- The Web Proxy knows the health of the UN and can route requests accordingly.
- Per User throttle for API Clients:
  - Default setting is 20 API req/sec per Active Unified Node. 4 Active Unified Nodes x 20 = 80 API req/sec (system wide) or 4800 API req/min (system wide). We recommend that you keep this setting.
  - This limit applies to all admin users, but in practice serves to limit API clients. Human admin users are not likely to create a traffic rate of 80 API req/sec.
- Self-service throttle. The default setting is 300 API req/min (per Unified Node). APIs for logins and actions would count against this throttle.

## 5.3. Configurable number of queue processes

### 5.3.1. Overview

#### Important:

- It is strongly advised that VOSS Support is consulted before making changes to the number of queue processes.
- The number of queues cannot be set to a value larger than the number of cores in the VM. A message Validate: `<num> is not a valid less_than_cores_number` will show in this case.

#### Available commands:

- `voss queues <number>` - Set the number of queue processes

This command restarts the voss-queue services.

`voss queues` - Get the number of queue processes

When using these commands, a CLI warning is shown to refer to this documentation:

Warning, updating this setting, without proper consideration of Best Practices or consultation with VOSS support, can lead to system instability.  
Do you wish to continue?

The number of queue processes is configurable in order to increase transaction throughput and will improve workload distribution across the cluster, but can only be made after considering other configuration changes or performance areas. These include:

- The maximum number of queues cannot exceed the number of cores in the VM
- Node memory configuration
- Impact on API and indirectly GUI responsiveness
- Number of workers for queue processes on different unified nodes
- Overall load on the primary node (node with primary database responsible for all database updates)

### 5.3.2. Node memory

When increasing queue processes, too little memory headroom can lead to out of memory errors on the unified node, which can cause services to be restarted and in rare situations, can also lead to database services being stopped.

The suggested required headroom per queue process should be considered as 4GB.

### 5.3.3. Impact on API and GUI responsiveness

A balance has to be created between the number of queue processes and API/GUI responsiveness. Increasing the number of queue processes on all nodes will increase the load on the primary node during high transaction load and the increased load on the database can lead to degraded API and GUI responsiveness if the number of queue processes are set too high.

### 5.3.4. Number of workers per queue process

---

**Note:** This consideration applies to the standard topology with unified nodes.

---

In order to alleviate load on the primary node, it is recommended to set the number of workers to zero. This will prevent any transactions from being processed on the primary node. This will allow the primary node to better service

- the higher query load from secondary nodes due to higher transaction load
- API requests requiring database interaction

A special consideration exists with setting the workers to zero on the designated primary node. When the primary node fails over to a secondary due to some event, the newly elected primary node will not have the number of queue workers set to zero, which could lead to an increased load on the newly elected primary that will process transactions, service API requests and service database queries.

Manual intervention will be required to set the number of workers to zero on the newly elected primary or restore the configured primary node to primary state.

It is recommended that monitoring be set up to automatically provide notifications in case of primary failover.

## 6. Onboarding customers and users

### 6.1. Planning guidelines for onboarding and ongoing operations

This is a high-level view:

- Number of Parallel operations, for example BL and QAS, for best performance:
  - BL: 4x500 (4 Bulk Load sheets in parallel with a maximum of 500 rows per sheet).
  - QAS: 5x200 (5 QAS Bulk Load sheets in parallel with a maximum of 200 rows per sheet).
- Recommendations for sync operations:
  - It is recommended that you schedule sync operations during off-peak hours.
  - During business hours, sync operations are slower due to the presence of other work on the system.
- Scheduler Template settings (20%, 50%, 80%):
  - For periods of high self-service and administrative work (including API clients), we recommend that the template is set to 20%.
  - For periods of moderate self-service and administrative work (including API clients), we recommend that the template is set to 50%. This is the value in the system prior to SU-1.
  - For periods of low self-service and administrative work (including API clients), set the template to 80%.
  - The current implementation only allows you to set 2 of the 3 values, that is a peak and an off-peak setting.
- In some cases, there are situations where Automate is used for changes. Ad Hoc syncs are best in this case.
- AS may choose to change Cisco UCM and sync back to Automate. This is where you must schedule daily off-peak sync operations.



## 7. Webex onboarding best practices

### 7.1. Add Automate over an existing Webex organization with configuration already in place

This procedure adds Automate over an existing Webex organization with configuration already in place.

Automate syncs in the configuration and automatically creates the required sites. Users, numbers, and devices are automatically moved to the appropriate sites.

**Note:** This scenario applies where a Webex organization has been previously configured in the Webex Control Hub, and Webex Calling configuration is in place (which includes locations, and may also include additional configuration; for example, for numbers, users, and devices).

1. In the Automate Admin portal, add the Automate customer.

The screenshot displays the 'New Record' form in the VOSS Automate Admin portal. The form is divided into two main sections: 'Customer Details' and 'Contact Information'.

**Customer Details:**

- Customer Name \*: Webex-calling
- Description: Webex Calling
- Extended Name: (empty)
- External Customer ID: (empty)
- Domain Name: (empty)
- Create Local Admin: ☒
- Cloned Admin Role \*: P1CustomerAdministrator
- Default Admin Role \*: Webex-callingCustomerAdministrator
- Default Admin Password \*: (masked with dots) ☐ Show
- Account ID: (empty)
- Deal IDs: (empty)
- Shared UC Applications: ☐
- Disable Number Management: ☐
- Public Sector: ☐
- Inactive Billing: ☐

**Contact Information:**

- Address 1: 100 Longwater Avenue
- Address 2: Green Park
- City: Reading
- State: Berkshire
- Postal Code: RG2 6GP
- Country: United Kingdom
- Name: (empty)
- Email Address: (empty)
- Telephone Number: (empty)

2. In the Automate **Global Settings**, configure the following:

- Disable HCS rules for the customer to allow Webex Calling number management without the restrictions of the Cisco HCS dialplan.

## 7.1. Add Automate over an existing Webex organization with configuration already in place

Global Settings

Number Inventory | Number Inventory Alerting | Webex App | Pexip Conference | Email | Phones | Voicemail | User | Flow Through Provisioning | Enabled Services | General Settings

Enforce HCS Dialplan Rules: No

Include the Number Inventory description in all number dropdowns: Inherit

Include the Number Inventory vendor in all number dropdowns: Inherit

Include the Number Inventory type in all number dropdowns: Inherit

Enable Number Inventory Cooling: Inherit

Number Inventory Cooling Duration (Days): 30

- Ensure that Webex App (Teams) is enabled for the customer to allow the display of conditional menu items.

Global Settings

Number Inventory | Number Inventory Alerting | Webex App | Pexip Conference | Email | Phones | Voicemail | User | Flow Through Provisioning | Enabled Services | General Settings

Enable Cisco CUCM: Inherit

Enable Cisco CUCC: Inherit

Enable Cisco WebEx: Inherit

Enable Cisco Webex App(Teams): Inherit

3. Optional. In Automate, create intermediate nodes, if required.

**Note:** This optional step allows sites to be grouped under intermediate nodes (divisions). This may be useful where there are a large number of sites and/or the administration of those sites should be available to groups of administrators, each of whom are responsible for a subset of those sites.

Webex-calling (Customer) VOSS Automate

Search for an action in the menu or dashboards

Hierarchy

Rows: 0 - 3 / 3

4 columns selected

Name	Description	Hierarchy node type	Located At
APAC	Asia Pacific	IntermediateNode	Webex-calling /Customer
EU	Europe	IntermediateNode	Webex-calling /Customer
USA	USA	IntermediateNode	Webex-calling /Customer

4. In Automate, create Webex Location Node mapping, if required.

**Note:** This step allows the mapping of partially matched location names to match to the intermediate nodes created in the previous step.

When syncing in locations from the Webex Control Hub, sites are automatically created under their appropriate intermediate nodes. Location names that don't match these rules are created under the Customer hierarchy.

## 7.1. Add Automate over an existing Webex organization with configuration already in place



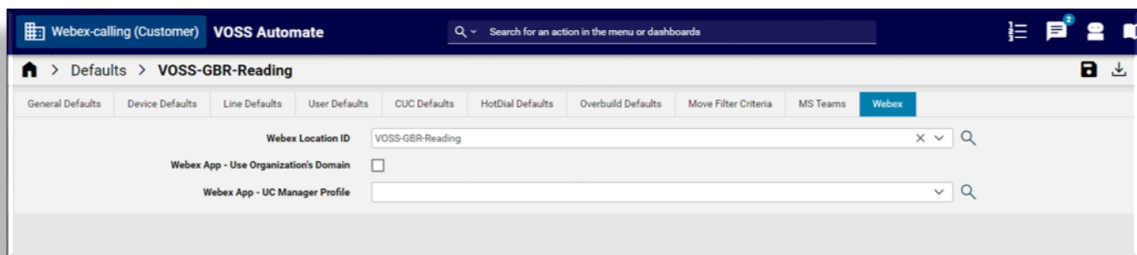
The screenshot shows the 'Webex Location Node Mapping' table in the VOSS Automate interface. The table has three columns: 'Intermediate Node', 'Search String', and 'Located At'. There are three rows of data, each representing a region (APAC, EU, USA) mapped to a specific VOSS location and the 'Webex-calling (Customer)' organization.

Intermediate Node	Search String	Located At
APAC	VOSS-AUS	Webex-calling (Customer)
EU	VOSS-GBR	Webex-calling (Customer)
USA	VOSS-USA	Webex-calling (Customer)

5. In Automate, in each site's *Site Defaults*, add or update the **Webex Location ID** field to prevent the automatic creation of new sites.

**Note:** In some cases, sites may already exist in Automate, typically, where Automate already provides managed services using different vendor technologies. In this case, update the existing Site Defaults Doc (SDD) parameter, **Webex Location ID**, for each site, prior to connecting Automate to the Webex Control Hub. Location names and Automate site names do *not* need to match.

Adding the *Webex Location ID* to the SDD prevents the creation of new sites.



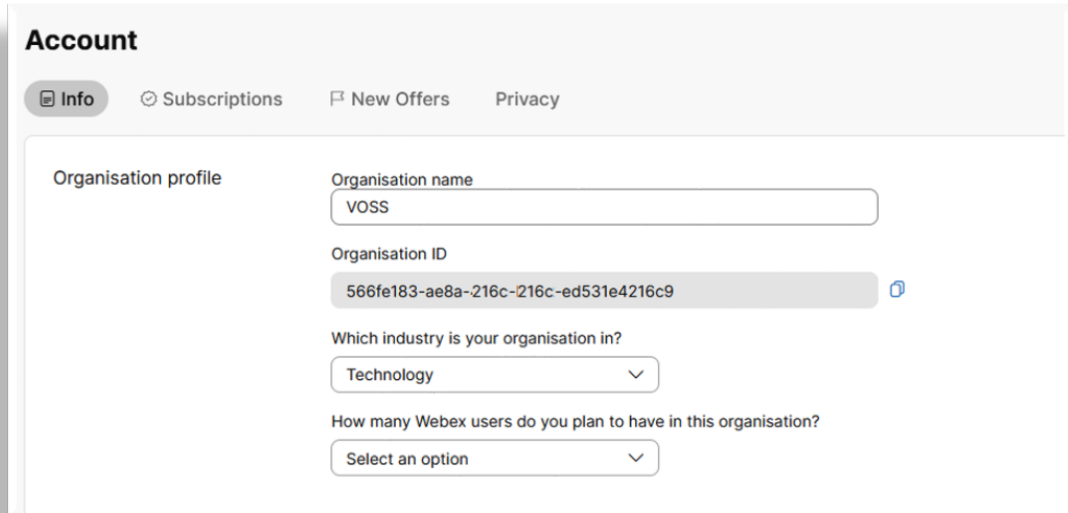
The screenshot shows the 'Defaults > VOSS-GBR-Reading' configuration page in the VOSS Automate interface. The 'Webex' tab is selected. The 'Webex Location ID' field is set to 'VOSS-GBR-Reading'. The 'Webex App - Use Organization's Domain' checkbox is unchecked. The 'Webex App - UC Manager Profile' field is empty.

6. Log in to the Webex Control Hub; then, configure the following:

- Obtain the **Organization ID** from the **Account** page.

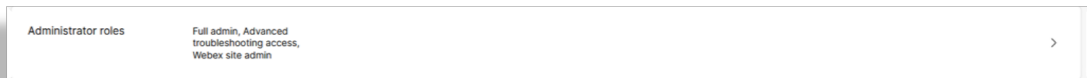
**Note:** The *Organization ID* will be required when connecting Automate to the Webex Control Hub.

## 7.1. Add Automate over an existing Webex organization with configuration already in place



The screenshot shows the 'Account' page with tabs for 'Info', 'Subscriptions', 'New Offers', and 'Privacy'. The 'Info' tab is selected, showing the 'Organisation profile' section. The 'Organisation name' is 'VOSS'. The 'Organisation ID' is '566fe183-ae8a-216c-216c-ed531e4216c9'. The 'Which industry is your organisation in?' dropdown is set to 'Technology'. The 'How many Webex users do you plan to have in this organisation?' dropdown is set to 'Select an option'.

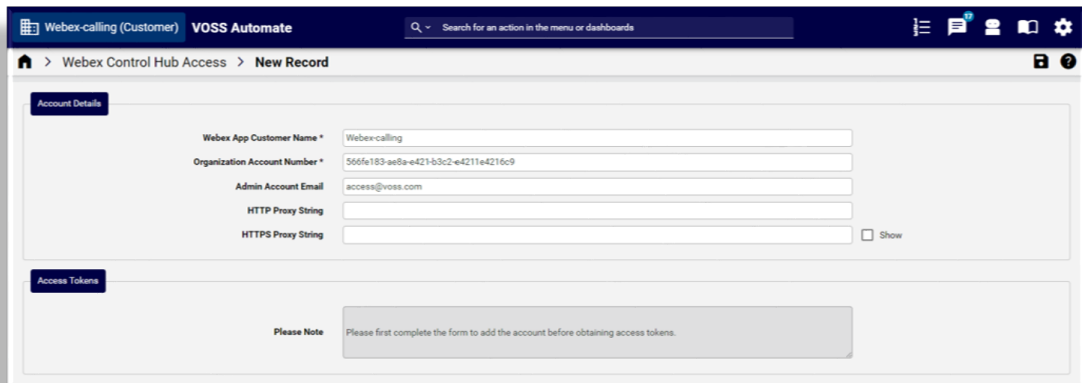
- Add an admin user that Automate can use to access the Webex Control Hub, and ensure that the correct roles are assigned.



Administrator roles	Full admin, Advanced troubleshooting access, Webex site admin	>
---------------------	---	---

### 7. In the Automate Admin Portal:

- Add the *Organization ID* you obtained from Webex Control Hub.
- If an HTTP(s) proxy is in use for internet access, fill out a value for **HTTP Proxy String** / **HTTPS Proxy String**.
- Save your changes.



The screenshot shows the 'Webex Automate' 'New Record' form. The 'Account Details' section includes fields for 'Webex App Customer Name \*' (Webex-calling), 'Organization Account Number \*' (566fe183-ae8a-e421-53c2-e4211e4216c9), 'Admin Account Email' (access@voss.com), 'HTTP Proxy String', and 'HTTPS Proxy String'. There is a 'Show' checkbox next to the proxy string fields. The 'Access Tokens' section has a 'Please Note' message: 'Please first complete the form to add the account before obtaining access tokens.'

- Re-open the form you saved, then click the **Connect to Webex Control Hub** link to open Webex Control Hub.

## 7.2. Add Automate management over an existing, un-provisioned Webex organization

The screenshot shows the 'Webex Control Hub Access' configuration page for 'Webex-calling'. It includes fields for 'Webex App Customer Name' (Webex-calling), 'Organization Account Number' (566fe183-ae8a-e421-b3c2-e4211e4216c9), 'Admin Account Email' (access@voss.com), 'Default Calling Behaviour' (a dropdown menu), 'Directory Synchronization Enabled' (checkbox), 'HTTP Proxy String', and 'HTTPS Proxy String'. There is a 'Show' checkbox next to the proxy strings. Below these fields is a 'Please Note' section with a link to retrieve or refresh tokens, and a 'Connect to Webex Control Hub' button.

- Log in to Webex Control Hub with the user credentials to be used for Control Hub access.
8. In Automate, execute the full sync to pull in all Control Hub data. Site-specific data is automatically moved to the sites created.

The screenshot shows the 'VOSS Automate' configuration page for 'SyncSparkWebex-calling'. It includes fields for 'Name' (SyncSparkWebex-calling), 'Description' (Syncs all Webex App device data for Webex-calling), 'Device Type' (Spark), 'Sync Type' (Pull from Device), 'Dependency Resolution' (a dropdown menu), 'Quick Import (Warning: Summary Data Only)' (checkbox), 'Import Schema Only' (checkbox), 'Import Data Only' (checkbox), 'Execute Asynchronously' (checkbox), 'Refresh Existing (Changed) Data' (checkbox), 'Force Refresh Of Data' (checkbox), 'Model Type List' (SparkDataAllMTL), 'Synchronization Order' (SparkDataSyncOrder), 'Model Instance Filter (Warning: Existing data not matching Filter will be removed)' (a dropdown menu), 'Device Filters' (a list with 'name: Webex-calling' and an 'Add Item' button), 'Workflows' (a list with an 'Add' button), and 'Transaction Log Level' (Warning). On the right side, there are three checkboxes under 'Disabled Operations': 'Disable Add Operation', 'Disable Update Operation', and 'Disable Remove Operation'.

The initial configuration and import of the Webex customer is now complete.

## 7.2. Add Automate management over an existing, un-provisioned Webex organization

This procedure adds Automate over an existing, un-provisioned Webex organization.

Automate will be used to create and manage Webex Calling configuration.

This scenario applies where a Webex organization has been previously created in the Webex Control Hub, but Webex Calling configuration does not yet exist.

**Note:** At the time of writing (25.1), some configuration must be managed via the Webex Control Hub. This topic does not include details for Webex setup outside of Automate, such as LDAP integration and *Organization* settings. Refer to the Webex documentation for these details.

1. In the Automate Admin portal, add the Automate customer.

The screenshot shows the 'New Record' form in the VOSS Automate Admin portal. The form is divided into two main sections: 'Customer Details' and 'Contact Information'. The 'Customer Details' section includes fields for Customer Name (Webex-calling), Description (Webex Calling), Extended Name, External Customer ID, Domain Name, Create Local Admin (checked), Cloned Admin Role (P1CustomerAdministrator), Default Admin Role (Webex-callingCustomerAdministrator), Default Admin Password (masked), Account ID, Deal IDs, Shared UC Applications, Disable Number Management, Public Sector, and Inactive Billing. The 'Contact Information' section includes fields for Address 1 (100 Longwater Avenue), Address 2 (Green Park), City (Reading), State (Berkshire), Postal Code (RG2 6GP), Country (United Kingdom), Name, Email Address, and Telephone Number.

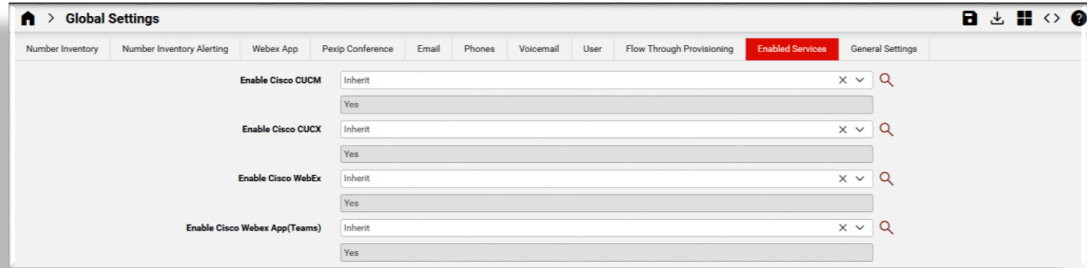
2. In the Automate Global Settings:

- Disable HCS rules for the customer to allow Webex Calling number management without the restrictions of the Cisco HCS dialplan.

The screenshot shows the 'Global Settings' page in the VOSS Automate Admin portal. The page has a tabbed interface with tabs for Number Inventory, Number Inventory Alerting, Webex App, Pexip Conference, Email, Phones, Voicemail, User, Flow Through Provisioning, Enabled Services, and General Settings. The 'Number Inventory' tab is selected. The settings include: Enforce HCS Dialplan Rules (No), Include the Number Inventory description in all number dropdowns (Inherit), Include the Number Inventory vendor in all number dropdowns (Inherit), Include the Number Inventory type in all number dropdowns (Inherit), Enable Number Inventory Cooling (No), and Number Inventory Cooling Duration (Days) (30).

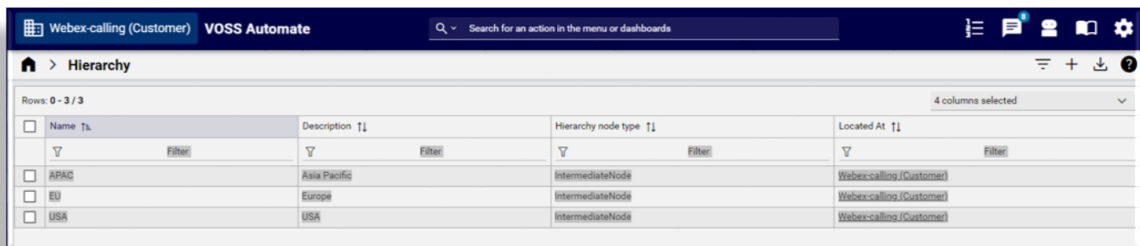
- Ensure that Webex App (Teams) is enabled for the customer to allow the display of conditional menu items.

## 7.2. Add Automate management over an existing, un-provisioned Webex organization

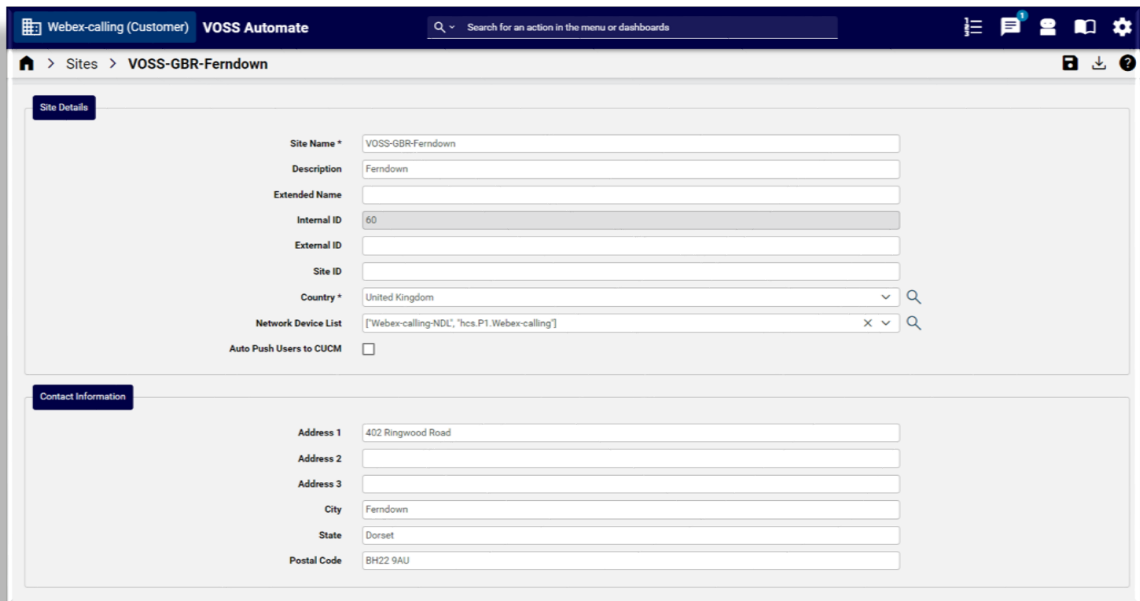


3. Optional. In Automate, create intermediate nodes, if required.

**Note:** This optional step allows sites to be grouped under intermediate nodes (divisions). This may be useful where there are a large number of sites and/or the administration of those sites should be available to groups of administrators, each of whom are responsible for a subset of those sites.



4. Manually add new Automate sites.

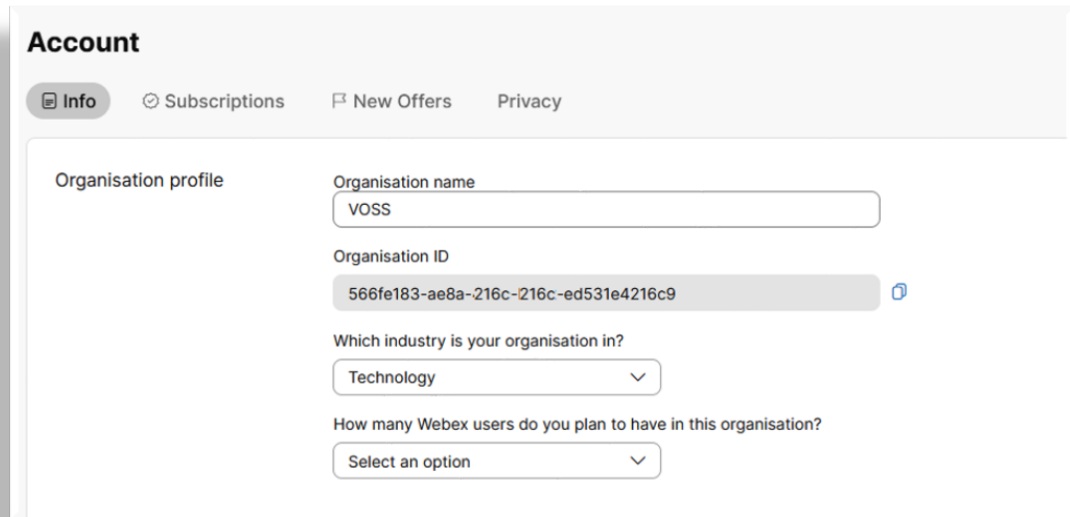


5. Log in to the Webex Control Hub; then:

- Obtain the **Organization ID** from the **Account** page.

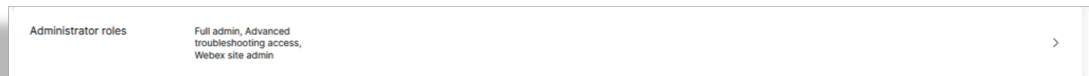
**Note:** The *Organization ID* will be required when connecting Automate to the Webex Control

Hub.



The screenshot shows the 'Account' page with tabs for 'Info', 'Subscriptions', 'New Offers', and 'Privacy'. The 'Info' tab is active, displaying the 'Organisation profile' section. The 'Organisation name' is 'VOSS'. The 'Organisation ID' is '566fe183-ae8a-216c-216c-ed531e4216c9'. The 'Which industry is your organisation in?' dropdown is set to 'Technology'. The 'How many Webex users do you plan to have in this organisation?' dropdown is set to 'Select an option'.

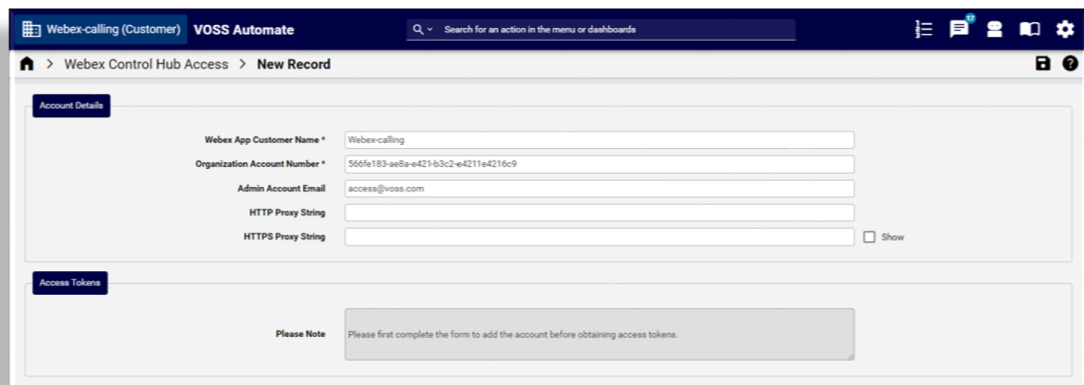
- Add an admin user that Automate can use to access the Webex Control Hub, and ensure that the correct roles are assigned.



Administrator roles	
Full admin, Advanced troubleshooting access, Webex site admin	

6. In the Automate Admin portal, add Webex Control Hub access:

- Add the *Organization ID* you obtained from Webex Control Hub.
- If an HTTP(s) proxy is in use for internet access, fill out a value for **HTTP Proxy String** / **HTTPS Proxy String**.
- Save your changes.



The screenshot shows the 'Webex Automate' interface with a 'New Record' form for 'Webex Control Hub Access'. The form has two sections: 'Account Details' and 'Access Tokens'. The 'Account Details' section contains fields for 'Webex App Customer Name' (Webex-calling), 'Organization Account Number' (566fe183-ae8a-e421b3c2-e4211e4216c9), 'Admin Account Email' (access@voss.com), 'HTTP Proxy String', and 'HTTPS Proxy String'. The 'Access Tokens' section has a 'Please Note' message: 'Please first complete the form to add the account before obtaining access tokens.'

- Re-open the form you saved, then click the **Connect to Webex Control Hub** link to open Webex Control Hub.



## 7.2. Add Automate management over an existing, un-provisioned Webex organization

The screenshot shows the 'Webex-calling' configuration page under 'Webex Control Hub Access'. It has two main sections: 'Account Details' and 'Access Tokens'.  
**Account Details:**  
- Webex App Customer Name: Webex-calling  
- Organization Account Number: 566fe183-ae8a-e421-b3c2-e4211e4216c9  
- Admin Account Email: access@voss.com  
- Default Calling Behaviour: (dropdown menu)  
- Directory Synchronization Enabled: ☐  
- HTTP Proxy String: (text field)  
- HTTPS Proxy String: (text field) with a 'Show' checkbox.  
**Access Tokens:**  
- A 'Please Note' box with the text: 'Please click on the link below to retrieve or refresh tokens.'  
- A link: 'Connect to Webex Control Hub'.

- Log in to Webex Control Hub with the user credentials to be used for Control Hub access.

7. In Automate, execute the full sync to pull in all Control Hub data.

The screenshot shows the 'SyncSparkWebex-calling' configuration page in the 'VOSS Automate' interface. The left pane is titled 'Details' and the right pane is titled 'Disabled Operations'.  
**Details:**  
- Name: SyncSparkWebex-calling  
- Description: Syncs all Webex App device data for Webex-calling  
- Device Type: Spark  
- Sync Type: Pull from Device  
- Dependency Resolution: (dropdown menu)  
- Quick Import (Warning: Summary Data Only): ☐  
- Import Schema Only: ☐  
- Import Data Only: ☐  
- Execute Asynchronously: ☒  
- Refresh Existing (Changed) Data: ☒  
- Force Refresh Of Data: ☐  
- Model Type List: SparkDataAllMTL  
- Synchronization Order: SparkDataSyncOrder  
- Model Instance Filter (Warning: Existing data not matching Filter will be removed): (dropdown menu)  
- Device Filters: A list with one item 'name: Webex-calling' and an 'Add Item' button.  
- Workflows: A button with a plus sign.  
- Transaction Log Level: Warning  
**Disabled Operations:**  
- Disable Add Operation: ☐  
- Disable Update Operation: ☐  
- Disable Remove Operation: ☐

8. In Automate, add locations to the Control Hub:

The screenshot shows the 'VOSS-GBR-Ferndown' location configuration page in the 'VOSS Automate' interface. The left pane is titled 'Details' and the right pane is titled 'Address'.  
**Details:**  
- Name: VOSS-GBR-Ferndown  
- Time Zone: Europe/London  
- Preferred Language: en\_gb  
- Latitude: (text field)  
- Longitude: (text field)  
- Notes: (text field)  
**Address:**  
- Address Line 1: 402 Ringwood Road  
- Address Line 2: (text field)  
- City: Ferndown  
- State: Dorset  
- Postal Code: BH22 9AU  
- Country: GB

9. In Webex Control Hub, set up calling connection:

- Go to **Locations**.
- Select the required location.
- On the **Calling** tab, add the PSTN connection type.

**Note:** This is reserved for future development in Automate.

10. In Automate, execute a full pull sync to pull location calling settings from Webex Control Hub into Automate. Location calling settings can now be managed from Automate.

The screenshot displays the VOSS Automate web interface. The top navigation bar includes the title 'Webex-calling : VOSS-GBR-Ferndown (Site)' and 'VOSS Automate', along with a search bar and various utility icons. The main content area is titled 'Webex Location Calling Details > VOSS-GBR-Ferndown'. It is divided into four panels:

- Details:** Contains fields for ID (Y2LzY29zcGFyzovL3VzL0xPQ0FUSU90LzNzY3OWFjLTRIMDC), Name (VOSS-GBR-Ferndown), Announcement Language (en\_us), External Caller ID Name (VOSS-GBR-Ferndown), User Limit (500000), PSTN Access Network Info, Outside Dial Digit, Routing Prefix, Default Domain (98027369.us10.bcid.webex.com), and Charge Number.
- Calling Line ID:** Contains fields for Name (VOSSDeviceDemo) and Main Number (None).
- Connection:** Contains a dropdown for Type (TRUNK) and a field for ID (Y2LzY29zcGFyzovL3VzL1RSVUSLLzZkZGExM2Y0LTA2NzUUNE).
- Music On Hold:** Contains checkboxes for Call Hold Enabled and Call Park Enabled (both checked), a Greeting dropdown (SYSTEM), and an 'Audio File' section with fields for ID, File Name, Media File Type, and Level.

11. In Automate, add numbers into the Automate inventory, and then push those numbers into the Control Hub.

The initial configuration and import of the Webex customer is now complete.

## 7.3. Webex syncs

Webex data should be synced from the Webex Control Hub, using schedules, in order to keep data up to date.

The SyncSparkXXXX data sync is automatically created, where XXX is the customer name. This sync should be scheduled to run at suitable intervals where it is interleaved with other Webex syncs.

A SyncSparkUsersXXXX data sync syncs only users, and is recommended where flow through provisioning is in use. This user sync may run more frequently than the full sync in order to reduce delays in provisioning or removing users.

In the case where numbers are updated frequently outside of Automate, a data sync to sync only numbers should run more frequently. This will be required not just for adding and deleting numbers, but also to update their usage status if numbers are allocated and de-allocated outside of Automate.

### 7.3.1. Webex Control Hub rate limiting

The Webex Control Hub may rate limit API calls in order to manage load on the Control Hub. Webex rate limit details are described at <https://developer.webex.com/docs/basics>.

In the event of rate limiting, the Control Hub will respond with a 429, and include a Retry-After header, defines the time that Automate should wait before retrying the API call.

It should be noted that the rate limiting is applied to all customers which were granted access via any one admin account. Therefore, a service provider using a single account to manage multiple customers will have the rate limit applied across all customers.

## 8. Data sync

### 8.1. General sync principles and best practices

#### 8.1.1. Sync overview

##### Overview

Automate provides several features for keeping the system in sync with underlying UC applications. This allows for the configuration and management of the UC apps outside of Automate, when required.

Sync feature	Description
Cache control policy	This Automate mechanism provides the ability to pull in the latest live data from the UC application(s) for the entity you're viewing, or at the time that it's needed - for example, before executing a change on that entity, to prevent any overwrite or setting conflict. Find out more about cache control policy behavior and configuration in the Data Sync chapter in the Core Feature Guide.
Data sync	This workflow pulls the latest data from the UC apps and updates the Automate cache when ran adhoc, or via a schedule. This is typically used for processes such as overbuild, to pull in the existing configuration from the UC applications or to pull in other changes made in the UC apps outside of Automate. For more information on the sync behavior and configuration see the Data Sync chapter in the Core Feature Guide.

---

**Note:** With the cache control policy in place, the need to setup and schedule sync regular syncs should aim to address any gaps that the cache control policy won't handle.

---

### When to use data syncs vs cache control policy

This section describes some prime use cases and guidelines for determining when a regular or scheduled sync might be required for an entity, instead of using the cache control policy.

---

**Note:** These scenarios assume some level of regular configuration being done to the UC apps directly, outside of Automate.

---

### Adding/removing entities in the UC applications directly

If you're adding or removing entities, such as users or phones, in the UC applications directly, then a sync is required to pull in new entities or to remove existing entities.

### Modifying key values that appear in the list views in Automate via the UC applications directly

If you're modifying key values that appear in the list views in Automate via the applications directly, such as changing a user's name, then an update sync may be required.

List view data is driven only from the Automate cache; thus, any updates made in the UC applications will not display in Automate until the entity is viewed in Automate (for example, opening that user), or when an update sync is run.

### Extracts run from Automate

Any type of extract that might be run from Automate, such as file dump, Automate Analytics, or billing feed, are based on cached data in Automate. Thus, a sync may be required if those capabilities are in use and any of the critical settings in those extracts are being managed outside of Automate.

### External clients accessing Automate via the API

External clients accessing Automate via the API have the `cached` flag available to request Automate cached data (`cached = true`), or to have Automate retrieve the latest data from the UC applications before responding (`cached = false`). Thus, the presence of this external client does not require a regular sync to be run as it can (and likely should) request the latest data in any case depending on the use case.

### Any other mods made on the entity

Any other mods made on an entity, such as call forward on a line via the CFwdALL softkey, will be pulled in when the record is viewed, and so won't necessarily require a sync. For example, if the only concern is that when executing an update to an entity, that the latest current settings are shown, then the cache control policy handles this without the need for a regularly scheduled update sync.

### Consider the Sync Hierarchy

When running a sync, manually or via schedule, the hierarchy (customer or site) at which you're running the sync determines where the items are pulled into. For example:

- A sync at the customer level pulls data in at customer level
- A sync at a site pulls data in at the site

For this reason, when setting up the sync, consider its purpose. If the items being pulled in need to be in a site, it may be more efficient to set the sync up at the site level, and run the sync at the site rather than syncing in at the customer level, and then having to move the various elements. This can even be done as a once-off sync, and using model type lists and model instance filters to grab the data relevant for the site.

## 8.1.2. Data sync types

### Overview

Automate provides the following data sync types:

Data sync type	Description
Pull from device	Available to all device types. <ul style="list-style-type: none"> <li>• Pull all data from the device</li> <li>• Pull only the schema from the device (used for LDAP)</li> <li>• Pull data from the Change Notification Feature local data collection</li> </ul>
Purge local resources	Available to all device types. <ul style="list-style-type: none"> <li>• Purge data from the cache</li> </ul>
Push to device	Available only to Cisco UCM devices <ul style="list-style-type: none"> <li>• Push data in the cache to the device</li> </ul>
Change notification sync	Available only to Cisco UCM devices

**Note:** A quick import option is available to fetch only summary data that is contained in a list operation response and not the data for all instances/fields. See Data Sync Overview in the Core Guide for details.

Generally, for all sync types, Automate builds up the lists of entities from both Automate and the device, and compares them, using the key for the device entity. The key is typically the unique identifier (ID) for the record in the device we're syncing with. For example, for Cisco UCM, the ID is the *pkid*, which is the internal Cisco UCM database ID.

For users, a sync builds up the list of `device/cucm/Users` in Automate and then requests from the Cisco UCM the lists of users it currently has for the comparison. Differences in the lists are handled according to each sync type.

## Related topics

- Data Sync Overview in the Core Feature Guide
- Change Notification Feature Overview in the Core Feature Guide

## Pull from device

For sync type *Pull from Device*, the Automate resource is updated where the same key is present in both lists. In this case, the device data is the master and the Automate system model data is updated with the device data.

For example, let's say new data is added to the Cisco UCM, so that the Automate system data state for a Cisco UCM device/*cucm/User* does not show instances that are shown on the Cisco UCM.

In this case, a *pull* data sync synchronizes the system data with the Cisco UCM data. For example, a user's Department may be updated on the Cisco UCM, but the update only shows on the system after a *Pull from Device* sync. If a user resource is created in Cisco UCM but not in Automate, this adds the device/*cucm/User* instance into Automate at the level the *pull* sync was run from, for example, at the customer level.

When deleting a Automate resource from the device, so that the key is in the Automate list but not in the device list, a *pull* sync removes the resource in Automate. For example, if the resource is a user in Automate but not in Cisco UCM, the *pull* sync removes the device/*cucm/User* record in Automate.

To restrict the number of records removed in Automate, ensure you have the following named macro at the hierarchy where the sync takes place:

```
PULL_SYNC_DELETE_THRESHOLD_<device_type>
```

For details, see Pull Sync Delete Threshold topic in the Advanced Configuration Guide.

When pulling device data, for example LDAP users from an LDAP device, the results returned to Automate depend on the LDAP server configuration. For example, if the returned results exceed the LDAP server configured maximum, and if the server does not support paging, an appropriate error message is returned.

For Microsoft 365 syncs, a **Max page size** (default 1000) setting can be adjusted if the error "Template output exceeded memory limit" is shown.

For details, see the Configure Microsoft Tenant Connection Parameters topic in the Core Guide.

## Push to Device

Sync type *Push to Device* is available only to Cisco UCM device types.

In a *Push to Device* sync type, devices are synchronized with the Automate system data state, which is the primary data state.

- When deleting device data from Automate so that the key is in the *device* list but not in the Automate list (for example, delete user in Automate), the user is removed from Cisco UCM. The user will not exist on the device or on Automate.
- When adding new device data to Automate so that the resource shows instances that are not shown on the device, a *push* data sync synchronizes the device data with the Automate data. For example, adding a device/*cucm/User* instance to Automate and running a *Push to Device* sync adds the user record to Cisco UCM.

Keys found in both lists are ignored. Existing records are not updated in either direction.

In the `device/cucm/User` example, if the same user exists on both Automate and on Cisco UCM, no update occurs in either direction. Detailed settings may still not match after a *Push to Device* sync.

---

**Important:** When performing a *push* sync, it is important to consider data dependencies between different models.

For example, data dependencies may exist between users and phones in the Cisco UCM. In this case, if a user is associated to a phone (via the associated devices on the user), you can't add the user if the phone does not yet exist in Cisco UCM.

On the other hand, for `ownerID` on the phone, pushing the phone first will fail since the user isn't in place.

This might mean running the *push* sync multiple times so it loads in the required order, or you may need to modify data (such as removing device association) to allow the *push* sync to succeed.

---

---

**Note:** The keys list sync logic described in this topic implies that in case of a reversion of the Cisco UCM to restores/inactive partitions, the end-state of the relevant pkids may differ to their state the last time Automate was in sync with Cisco UCM (before a restore), particularly if testing occurred in between. This means you may, for example, have a user with the same username in both Automate and Cisco UCM, but if that user's pkid in Cisco UCM now differs to the one in Automate from previous syncs or interactions, they will be seen as different users even though they have the same usernames.

---

## Change Notification Sync

Sync type *Change Notification Sync* is available only to Cisco UCM device types.

A *Change Notification Sync* is a pull sync of changes stored in the local collection that is updated by the Change Notification Collector service.

For more details on Change Notification Sync, see the related topics in Data Sync section of the Core Feature Guide.

## Purge Local Resources

In a *Purge Local Resources* sync type, all resources or instances of device information that exists in the system are deleted. Entities in the device are not deleted.

---

**Note:** The default *purge* syncs created when adding a Cisco UCM, CUC, LDAP or CCX server type are disabled by default. To use the *purge* sync, the "Remove" check box must first be cleared on the "Disabled Operations" tab of the relevant sync.

---

This sync type is typically used when cleaning up the system. The system displays a warning before executing an enabled *purge* sync.

See the following sample device type syncs:

- `HcsPurge-{{CUCMHostname}}-{{CUCMClusterName}}-DS`
- `HcsUserPurgeDS-{{CUCMHostname}}-{{CUCMClusterName}}`
- `HcsPhonePurgeDS-{{CUCMHostname}}-{{CUCMClusterName}}`
- `HcsPurge-{{CUCXHostname}}-{{CUCXClusterName}}-DS`



- PurgeUccx-{{UCCXHostName}}
- HcsLdapUserPurge-{{UniqueID}}
- PurgeSpark{{CustomerName}}

### 8.1.3. Scheduling syncs

- When scheduling syncs, avoid too many overlapping syncs at a given time. Automate already blocks multiple syncs against a given device.
- The best practice is to not have more than five syncs running at a given time.
- To avoid load and issues with concurrency, schedule syncs carefully and at intervals when they are really required. For example, do not run nightly syncs unless it is necessary. Since syncs generally cover the case where information is changed on the UC apps outside Automate, the level of third party integration or direct configuration tasks should play a role in the decision to schedule. For details, refer to the topics on Cisco UCM, CUC and LDAP below.
- Since it is possible to limit the number of records processed with a given sync, more predictability can be obtained with scheduling.

## 8.2. Cisco UCM

### 8.2.1. Cisco UCM sync

Cisco Unified Communications Manager (Cisco UCM) supports two types of sync:

- Regular API sync - utilizing the regular use of LIST and GET API calls to retrieve data; like any other device sync.
- Change notification sync - utilizes a service on the UCM side to pull information about records that have changed in a given period. Note: Model Instance Filters cannot be used in conjunction with a change notification sync.

The change notification sync type is generally the most efficient sync type to use, as it minimizes the amount of data that needs to be retrieved from the UCM (especially for updates).

The change notification sync process works as follows:

Automate retrieves the change records from the UCM on a regular interval (configurable). For example, this could be every 10 minutes. At the time of a scheduled sync is setup, Automate processes the change records collected (for example, nightly). Automate then processes the records accordingly:

- Add - will do a GET API call to retrieve the full record and add it to Automate.
- Update - will do a GET API call to retrieve the full record and update the record in Automate.
- Del - will remove the record from Automate.

So the efficiency on update syncs is because we do not need to do a GET API call for every single record in the system - only those that changed. In large UC application installations, this can make a big difference in Update sync times.

## 8.2.2. Update sync operations

Partners can use Cisco UCM's *Change Notification* functionality to process *update sync* operations faster. By default, this feature is disabled, but can be enabled by the partner.

This topic provides guidelines for setting up a sync schedule, and lists associated performance implications. Further details for using *Change Notification* is provided elsewhere in the Automate documentation. The changes described here are not transactions. For this reason, details don't display in the transaction log; instead, these display in special logs created for this feature.

These guidelines are derived from concepts related to total processing capacity. The total number of updates processed in a time period is the sum of all updates across the customers selected for update in that time period. In this case, the time period is one hour. In this example, it is assumed that each customer has 1000 UCM-related changes in that hour.

Recommendations provided in the following table indicates that 5 customers can run in parallel (concurrently), and therefore a total of 5,000 changes processed in total. Partners exceeding the recommendation of 5 concurrent customers may notice a performance degradation, and the full set of required changes may not complete within that hour. Alternatively, if the number of changes for any customer is significantly higher than 1,000, or if the total number of changes is significantly greater than 5,000, the supported concurrency number may be less than 5.

If some of the planned changes do not complete within the hour noted in the table, those changes are completed the next time that particular customer is scheduled for a sync.

If the number of changes for any customer is so large that the changes continually exceed those that can be processed in one hour, it will eventually result in a full sync. For such customers, it is recommended to schedule within an hour where less than 5 customers execute concurrently.

Configuration	Recommendation
Maximum number of concurrent CNF sync	5
Maximum number of changes processed per CNF sync	1,000
CNF sync schedule frequency	Once per hour per customer - This is subject to the staggering of CNF sync across customers.
Staggering of CNF syncs across customers	Factor of maximum changes processed and maximum number of concurrent CNF syncs.
CNF collector frequency	Initial recommendation is 15 minutes.
When is full sync required?	Weekends only or when there are CNF alerts prompting for full sync.

**Note:** It is recommended that you disable the functionality if you experience significant performance degradation. Contact your support representative if you have any performance concerns.

### 8.2.3. Staggering CNF syncs across customers

This section provides an example and considerations for change notification (CNF) syncs across customers. If a partner has twenty customers who want to use CNF sync, only schedule a maximum of five CNF syncs to run concurrently. This means that syncs would run as follows:

- 1st hour, for example 12:00  
Customer 1 to Customer 5
- 2nd hour, for example 13:00  
Customer 6 to Customer 10
- 3rd hour, for example 14:00  
Customer 11 to Customer 15
- 4th hour, for example 15:00  
Customer 16 to Customer 20
- 5th hour, for example 16:00 (Begin repeating customers)  
Customer 1 - Customer 5
- and so on.

The preceding example means that the CNF sync schedule per customer must run at 4 hour intervals. Therefore, there are 6 CNF syncs per customer within a 24 hour window. With each CNF sync processing up to 1k changes, there are:

- A total of 6k changes processed per customer in a 24 hour window
- A total of 120k changes processed across all 20 customers in a 24 hour window

### 8.2.4. Recommended Cisco UCM sync setups

#### Overview

This section provides Cisco UCM sync recommendations.

#### Bottom-up user sync

If using bottom-up sync into Cisco UCM, the users are added to UCM via LDAP. In this scenario they do not appear in Automate in order to be managed until they're synced in.

**Note:** If you use this sync in a multi-cluster environment, additional guidance on the user sync setup is provided in the Multi-Cluster Deployments Technical Guide.

- Recommended setup
  - Model Type List - device/cucm/User
  - Actions - Add/Update/Del all enabled.

- When to use - scheduled. The most frequent this should run is in line with the LDAP->UCM sync time (typically once every 24hrs but minimum of every 6hrs or so). The length of this sync should consider the maximum allowable time for an end user to be in the system in typical business practices. Edge cases can always be handled in between scheduled syncs by running the sync manually if required - that is often better than having a very frequent sync that is not typically needed.
- Events - the different actions (add/update/del) have different post execution events for the device/cucm/User model type that need to occur. These handle various aspects of the user setup. See below for a screenshot of the setup for an example:
  - \* Add Operation workflow = UserCucmSyncAdd
  - \* Update Operation workflow = UserCucmSyncUpdate
  - \* Delete Operation workflow = UserCucmSyncRemove
- Change notification should be used for this sync to manage load (except if using a model instance filter).

Workflows fields of the event setup on the Cisco UCM user sync:

- **Model Type:** device/cucm/User  
**Operation:** Add  
**Phase:** Post Execution  
**Workflow:** UserCucmSyncAdd
- **Model Type:** device/cucm/User  
**Operation:** Update  
**Phase:** Post Execution  
**Workflow:** UserCucmSyncUpdate
- **Model Type:** device/cucm/User  
**Operation:** Delete  
**Phase:** Post Execution  
**Workflow:** UserCucmSyncRemove

### Phone types and related entities

This will force Automate to retrieve the latest phone type data from the UCM and related entities like phone button templates, and so on. This is not possible via the change notification in Cisco UCM today.

- Recommended setup:
  - Model type list including: *device/cucm/PhoneType*, *device/cucm/PhoneTemplate*, *device/cucm/securityProfiles*
  - Actions - Add/Update/Del all enabled
  - When to use - Not scheduled - run ad hoc as needed. This includes post UCM upgrades, installation of a new device COP file in UCM, managing phone button templates, managing device security profiles. If you are not seeing a phone type of the button template in Automate that you are expecting, running this sync will likely resolve it.

## Other syncs

Additional syncs can be set up for specific requirements, based on the implementation.

**Important:** In setting up processes that sync any new entities into Automate, these will add the entities to the hierarchy level of the sync. So this will require the use of overbuild or ad hoc move processes to get the entities into the right site, for example, if needed (such as users, phones, lines, and so on).

## 8.3. Cisco Unity Connection

### 8.3.1. CUC sync

User-related services such as *unified messaging* or *alternate extension* are only imported when the user is added or updated (for example, when updating their first name, last name, or email address).

- If services are added directly to the user on CUC, for example, when adding Unified Messaging or user-related services such as Alternate Extension, this service will not be imported when running the next full import from CUC. To import these services a model type list must be applied to a dedicated Data Sync to target the required model types. A default Model Type list **CUCXN Overbuild Resources** exists for this purpose, which includes the following model types:
  - device/cuc/User
  - device/cuc/UserPassword
  - device/cuc/UserPin
  - device/cuc/AlternateExtension
  - device/cuc/SntpDevice
  - device/cuc/SmsDevice
  - device/cuc/PagerDevice
  - device/cuc/PhoneDevice
  - device/cuc/HtmlDevice
  - device/cuc/Callhandler
  - device/cuc/CallhandlerMenuEntry
  - device/cuc/CallhandlerTransferOption
  - device/cuc/Greeting
  - device/cuc/MessageHandler
  - device/cuc/ExternalService
  - device/cuc/ExternalServiceAccount
- If making changes on the CUC directly to the schedules, then it is recommended that a dedicated sync be created which will pull in the all the Schedule related models (4 models) using the MTL **CUCXN Schedules**.

- It is recommended to use the CUCXN Exclude ImportUser MTL on CUC data syncs in order to avoid unnecessary data and slowing the sync time.

## 8.4. LDAP

### 8.4.1. LDAP sync

The LDAP sync process currently only supports regular syncs.

## 8.5. Cisco Webex App (Spark)

### 8.5.1. Cisco Webex App sync

If Cisco Webex App (Spark) is part of the solution and being managed, there are a number of considerations around sync with Cisco Webex App.

The typical setup is that the Cisco Webex App users are fully managed by the Automate system so there is no need for user sync. In this setup the only sync required is to pull in basic system data from Webex App for Automate to utilize in user configuration. A sync for this is added into Automate when a Cisco Webex App Service is added to the system and is executed automatically after Service creation or can be initiated by an admin as needed:

- SyncSparkRolesLicenses<customername> - Sync of basic data - e.g. licenses and roles, etc.

In an alternate scenario where some element of user management is occurring outside of Automate (for example, LDAP Connector), then a user sync will be required to pull that data into Automate for further management. Once the users are synced into Automate, they need to be moved to the appropriate site with the rest of the end user's services to be further configured and managed. This move can be done via the Webex App menu item by selecting the users and then using the **Action > Move** option to move them.

This sync can be initiated by an administrator as needed or if required, a schedule can be setup to run the sync on a regular interval.

- SyncSpark<customername> - Full sync of Webex App (Spark) including user data.

When Automate is integrated with a customer's user directory, the normal user management approach applies, in other words:

- Users are synced into Automate at the Customer hierarchy level
- Users must be moved to the relevant Site hierarchy level
- Once at the correct Site level, Quick Add User or Advanced User can be used to enable services (Webex App in this case) for the users

## 8.6. Microsoft Graph and Teams sync

### 8.6.1. Microsoft syncs

#### Overview

This topic provides an approach to syncs with “auto filtering” for Microsoft.

#### Using filters in syncs

It is recommended that you define a filtering strategy employing filtering criteria that can drive filtering on the Microsoft Graph API side.

Applying the filter on the Microsoft side in this way returns matching users only, thus minimizing the number of records that need to be processed, and reducing sync time and load on the system. For example, let's say there are three hundred thousand users in your tenant and you only need fifty thousand users in the system to be managed. In this case, it is recommended that you apply API filtering to retrieve only the relevant fifty thousand users, while preventing import of the additional two hundred and fifty thousand records that you don't need.

#### Supported fields

The following fields are supported for Microsoft Graph API filtering:

- UPN
- city
- country
- IsLicensed
- Licenses
- UserType
- CompanyName
- employeetype
- department
- office
- ExtensionAttributes(1-15)

---

#### Note:

- These fields only support the *equals* criteria. Currently, any other criteria can't be applied via the API call.
- For *IsLicensed*, the parameter for *equals* must be *true*, and must be lower case to match. Any other value for *equals* is treated as *false*.
- For *UserType*, the parameter for *equals* is *Member* by default so that *external/Guest accounts* are excluded.

Building your MsolUser model instance filter (MIF) with these fields and using the required criteria ensures that you're using the API filtering in Microsoft Graph where possible.

---

**Note:** Filtering on the Microsoft Graph API is case-insensitive, which helps to prevent errors where, for example, User Principle Names (UPNs) are not uniform.

---

If filtering on a field or using criteria that is unsupported on the Microsoft Graph API, then this is applied within Automate after retrieving all records from Microsoft.

### Filtering methods

There are two approaches to filtering:

- *Auto filter - filter CsOnlineUser based on MsolUser*
- *Traditional syncs - independent filters*

#### Auto filter - filter CsOnlineUser based on MsolUser

The *Auto-filter feature for Teams/CsOnlineUser sync* filters CsOnlineUser based on MsolUser. Auto filter is the recommended default if filtering is used. Auto filter simplifies setup and operations and should be reviewed as a path to move to for existing customers/systems that might have implemented other filtering approaches prior to Automate 25.1.

Auto filter provides the ability to focus filtering around the Microsoft Entra User (MsolUser) resources, which means that the system only pulls users from Teams (CsOnlineUser) if the user exists in Automate (filtering based on UPNs). As a result, you won't sync in any Teams users that haven't first been synced from Microsoft Entra.

Sync setup and performance is streamlined, making it easier for admins to set up and manage filters on the syncs. Admins can build effective filters for various needs, based on the rich set of fields available on Microsoft Entra User. The Teams sync utilizes this data set rather than another filter, and excludes any non-essential data. Syncing with auto filter impacts only the `device/mstteamsonline/CsOnlineUser` model. No other Teams models are affected.

Model instance filters and related filtering capabilities are supported for all other Microsoft Teams models, but not *CsOnlineUser*.

To use the auto filter, you apply filters to the O365/MSOL user sync, which has more fields to filter on, and use the *Auto-filter feature for Teams/CsOnlineUser sync*. This will *not* support additional filters on the CsOnlineUser.

The table describes the advantages and disadvantages of the *Auto-filter feature for Teams/CsOnlineUser sync*:



Advantages	Disadvantages
<ul style="list-style-type: none"> <li>• Focuses filter activity on the user model that has the most fields to filter on -</li> <li>• Easy to create a plan to “stamp” users if the base populated fields are inappropriate. For example, end customers can “stamp” users for managed services to their MSP.</li> <li>• Provides significant control for managing the dataset size for various reasons, such as performance or security (to only see relevant data, for instance)</li> <li>• Provides a more effective method for limiting the Teams sync and improving sync performance (not limited to filtering based on CsOnlineUser fields)</li> </ul>	<ul style="list-style-type: none"> <li>• It is not possible to apply additional CsOnlineUser filters to the sync if you’re using the <i>auto filter</i>. This is due to how filtering happens on the PowerShell side, and the risk of unpredictable results.</li> </ul>

### Traditional syncs - independent filters

Traditional syncs (using independent filters) refer to the way that syncs have typically been set up and managed in Automate. In this method, the sync is set up with separate model instance filters for the Microsoft Entra and Teams side, and you would try to align these model instance filters where possible. This sync works when a field you want to filter on is the same on each model (for example, *City* or *Country*), but are often not the ideal fields to filter on.

Advantages	Disadvantages
Can offer more control in scenarios where there is a need to have different sets of users from Microsoft Entra compared to Teams.	<ul style="list-style-type: none"> <li>• Difficult for admins to set up, to ensure accuracy, and to maintain</li> <li>• Inefficient performance Many filters need to be applied on Automate, which means all users are pulled then filtered.</li> <li>• Often requires trade-offs (pulling in too much data due to lack of effective fields to filter on)</li> </ul>

### Filtering and user types

The following user types should be considered when defining filters, regardless of the approach you take:

- Real users
- Service accounts

The table describes filtering considerations for the user types:

User type	Considerations
Real users	<p>Real users need to be in MS Entra to have Teams, so should not be an issue to have the user come in via Entra and identify an appropriate filter.</p> <p>The goal is to minimize the users being pulled in for performance/security reasons, so identifying a filter based on your data to best achieve this.</p> <p>This may be considered the base filter.</p>
Service accounts	<p>The potential challenge with this user type is how these users fit into the MsolUser filter.</p> <ul style="list-style-type: none"> <li>• <b>Resource accounts</b> - these often won't have the same field(s) populated as real users (for example, City), as they are auto-created and not effectively managed. Criteria therefore needs to be added to the filter to incorporate these if they don't fit into the "real user" filter. At the highest level, these can be filtered on the <i>Department</i> field to be synced in. This means all resource accounts are pulled in instead of a potential relevant sub set. However, if the Entra users for these accounts are tagged correctly then the filter can be refined.</li> <li>If all resource accounts are pulled in, this is typically a far smaller set of data and less sensitive than real users so the lack of potential filtering is probably acceptable considering the admin effort of tagging those Entra users correctly.</li> <li>• <b>Meeting Room and Common Area Phone users</b> - These instances will also have Entra users for each instance. Typically, these are more like "real users" as they need to be explicitly added, they usually have a site (location), and other business context that can be added in Entra. Thus, if they align with the "real users" filter, nothing further is required. However if they don't align with "real users", your base filter may need to be extended to ensure these users are encapsulated.</li> </ul> <hr/> <p><b>Important:</b> The CsOnlineUser record for service accounts are pulled in from Teams to enable accurate inventory and management of the user accounts.</p> <hr/>

### MsolUser filter logic in the model instance filter

Your MsolUser filter logic in the model instance filter (MIF) should consider the following:

- Criteria required for "real users" based on the fields supported on MsolUser.

This can include fields populated naturally in the directory (such as City or Country), or by using a specific tagging approach (such as *ExtensionAttribute1 = Managed*)

- Criteria for service accounts (resource accounts, meeting rooms, and common area phones)

Where these can align to the "real user" filter, this is preferred. However, if alignment is not possible, ad-

ditional criteria may need to be added to the filter (for example, *Department = Microsoft Communication Application Instance*).

### Recommended setup

This section describes the recommended, baseline setup to get started with your sync scenarios, and may be applied in various scenarios, for example, Enterprise with a tenant, or Provider with multiple end customers. You can adjust the recommendations as needed.

Setup recommendations are provided for:

- MS365 syncs
- Teams syncs

### MS365 syncs

MS365 syncs focus on the resources that the system pulls via the MS Graph API, including tenant resources, such as Entra users, licenses, or groups, and limited Teams elements.

This section will give some overall guidance on the suggested baseline sync setup and considerations.

### User filtering (MsolUser)

Unless you really need to see all Entra users in the system or the user base is very small, then some filtering should always be applied to the syncs for Entra users. This is to minimize the time and resources spent on syncing as well as to improve security as only relevant user data is in the system.

Common scenarios to consider in your Model Instance filters (MIF) for MsolUser (Entra Users) to pull into the Automate system:

- If you're handling Microsoft licensing outside of the Automate system, or if you're only applying incremental licenses, such as Calling Plan, MCOEV, or extra group licensing:
  - Apply the *isLicensed = true* option
 

This option only syncs in users that have some sort of license assigned from Entra. Users without any licenses are excluded.
  - If you intend to use Automate to fully license users from scratch, then leave out this filter so that users without licenses are synced in to be visible for provisioning.
- Additional criteria to match users required - some examples:
  - Specific license(s) - for example, E5 or MCOEV
  - City(s)
  - Country(s)
  - Field identified to be used for filtering - ExtensionAttribute1, Custom Attribute, other schema field - populated with some specific value, and so on.
- System Accounts
  - Resource accounts - if the previous filters won't match your resource accounts, then adding additional filters for this. For instance, *Department = Microsoft Communication Application Instance*

- Meeting Room/Common Area Devices - attempt to match your additional criteria for users above, but if not, include any additional criteria to identify these (could be by specific license, for instance).

**Note:**

- A default filter is applied on the API to only sync in users with *UserType* = 'Member' - this excludes external/guest accounts from the sync.
- Consider the length of your filter when using fields that are filtered on the API (recommended). There is no documentation from Microsoft on the maximum length of the filter so testing will be key to ensuring your filter doesn't cause an error from Microsoft (typically indicates invalid filter in error message). If you need a long filter and it causes an error on the Graph API, you will need to break your MIF up into multiple syncs. For this reason, it's recommended that you avoid very long MIFs that have multiple different criteria, if possible.

If using multiple criteria, ensure you match the logic needed - *AND* conditions compared to *OR* conditions. Multiple top-level criteria are treated as an *OR* condition where multiple inner level criteria are treated as *AND*.

You can also set up multiple syncs around users to have the criteria separate, if needed. In this case, carefully test the *remove* and *purge* users that don't match the settings on the sync to ensure the behavior is as expected. The "auto-filter" capability will work even if you have multiple different user syncs set up around users.

**Note:** Refer to the documentation on model instance filters for more information.

### Recommended sync setup for MS365

The system automatically creates a *Full* and *User* sync variant. It is recommended that you ignore these syncs and instead set up the specific syncs needed for your setup. Auto-created syncs will reappear if you delete them and can potentially be updated during upgrades.

The table describes the recommended syncs to build as a baseline, and for scheduling purposes:

Sync	Description
M365 Sync Excluding users	Use this sync for all elements except users as that will be handled via User sync. This can be run on a longer schedule (for example, once a week or month), or even adhoc when needed (for example, a new license is added to the system).
M365 Sync User Add/Remove	This will be the main sync setup that runs regularly to bring in new users or to remove users offboarded outside the system. As this focuses on just adding and removing users it keeps the sync load and runtime minimal.

**Note:** A sync that updates users is not been included since, typically this does not need to be scheduled. When opening users in the system, the latest settings are pulled in from Microsoft so it can be viewed that way. If you find a need to run update syncs regularly (for example, you want to utilize flow through provisioning to move users between sites based on updates, or many changes are being made outside the system), this can be set up and scheduled as needed. However, note that update scenarios are the most time-consuming in the sync as each individual record needs to be compared for changes.

Additional syncs may be set up for adhoc administration use. In this case, ensure they have similar settings and the appropriate Model Instance Filter (MIF) if the sync involves users.

### M365 Sync (excluding Users or excluding Users and Groups/Teams)

This sync handles elements such as licenses, Teams, and Groups. The main purpose would be to bring in new instances or to remove old instances of these elements. Updates can be handled by opening the entity in the Admin Portal if or when the entity is managed. If there are additional elements that you don't want to sync in, you can build an appropriate Model Type List for those elements. However, the result should at least be that MsolUser is not part of this sync.

The table describes the settings to use:

**Note:** You can clone the auto-created M365 sync as a starting point.

Setting	Description
Model Type List	This should be the list of elements you want to exclude, and should be at least Excluding Users OR alternative Model Type List
Refresh Existing	Set to <i>true</i> (checked)
Model instance filter	Unlikely to be required unless for some specialized need.
Disabled Actions	Update <i>true</i> (checked)
Purge Unmatched Records	<i>true</i> (checked)

### M365 sync user Add/Remove:

This sync pulls in new users and removes users that no longer match the filter (or were removed from Entra AD).

The table describes the settings to use:

**Note:** You can clone the auto-created M365 user sync as a starting point.

Setting	Description
Model Type List	M365 User Data (with just msgraph/MsolUser in it)
Refresh Existing	Set to <i>true</i>
Model instance filter	The user MIF you set up with your user filter criteria
Include the standard user work-flows	
Disabled Actions	Update <i>true</i> (checked). Note that in this case, you will need to uncheck <i>Remove</i> .
Purge Unmatched Records	Set to <i>true</i> (checked)

## Teams

These syncs focus on the resources that the system pulls via PowerShell from MS Teams, that is, most Teams and voice elements. This section provides overall guidance on the recommended baseline sync setup and considerations.

### User filtering

For user filtering, it is recommended that you use the *auto-filter* that will automatically only pull Teams users that the system has a corresponding MsolUser record for. This enables you to use filtering capabilities available on the MS 365 sync (see above) to also filter the Teams users. In this case, there is no additional filtering required.

If you're not using the *auto-filter*, then, as for the MS365 sync, you will need to define an appropriate model instance filter to match the Teams users to sync in. This should only be done if you require different filter criteria for Teams than you have for Entra users.

**Note:** The system automatically applies an additional filter over the *auto-filter* or Model Instance Filter (MIF) that is configured:

- *AccountType*, for example, User or ResourceAccount - this ensures only these account types are synced in, which filters out elements such as external/guest accounts

And;

- *softdeletiontimestamp=null* - this eliminates the accounts that have been removed/deleted but still exist in Teams as 'soft deleted'.

### Recommended sync setup for Teams

The system automatically creates a *Full* and *User* sync variant. It is recommended that you ignore these sync variants and load/set up the specific syncs required for your setup. The auto-created syncs reappear if you delete them and can potentially be updated during upgrades. For this reason, they should be ignored.

The table describes the recommended syncs to build as a baseline, and for scheduling purposes:

Sync	Description
Teams Sync Excluding users	This will be the sync to use for all elements except users as that will be handled via User sync. This can be run on a longer schedule (for example, once a week or month) or even adhoc when needed.
Teams Sync User Add/Remove	This is the main sync setup that runs regularly to bring in new users or to remove users offboarded outside of the system. As this focuses on just adding and removing users it keeps the sync load and runtime minimal.

**Note:**

- A sync that updates users has not been included as typically this doesn't need to be scheduled. When opening users in the system, the latest settings are synced in from Microsoft so it can be viewed that way. If there is a need to run update syncs regularly (for example, when many changes are being made outside the system), this can be set up and scheduled as needed. However, update scenarios are most time-consuming in a sync as each individual record needs to be compared for changes.

- You can set up additional syncs for adhoc administration use, if required. In this case, ensure they have similar settings and the appropriate Model Instance Filter (MIF) if the sync involves users.
- For planning, (if not using *auto filter*), if you're using fields that can be filtered on the PowerShell side (for example, *City*), there is a limit to the length of filter that can be applied. This is an undocumented PowerShell limitation, and failure messages may be unclear (for example, they may refer to relevant messages about the filter to a message containing "Expected literal (number, boolean, or null)". Internal system testing indicate 97 conditions for a single field is the limit. For instance, 97 cities passed, whereas 98 cities caused a failure; 97 cities and 97 countries passed and 98 of either failed. For this reason, it is recommended that you test and validate scenarios for your tenant if you're going to require long filters.

### Teams sync excluding users

This sync handles all Teams elements, except users. The main purpose of this sync is to sync in new instances or to remove old instances of these elements (updates can be handled by opening the entity in the Admin Portal if/when they manage it). If there are additional elements that you don't want to sync in, you can build an appropriate Model Type List for this purpose. However the result should at least be that CsOnlineUser is excluded from this sync.

The table describes the settings to use:

**Note:** You can clone the auto-created Teams sync as a starting point.

Setting	Description
Model Type List	To define the list of elements that should be excluded; should be at least Excluding Users OR alternative Model Type List
Refresh Existing	Set to <i>true</i> (checked)
Model instance filter	Unlikely to be required, unless for some specialized need.
Disabled Actions	Update <i>true</i> (checked)
Purge Unmatched Records	Set to <i>true</i> (checked)

### Teams sync user add/remove

This syncs in new users and removes users that no longer match the filter (or were removed from Teams).

The table describes the settings to use:

**Note:** You can clone the auto-created Teams user sync as a starting point.

Setting	Description
Model Type List	Teams user data (contains just msteamsonline/CsOnlineUser)
Refresh Existing	Set to <i>true</i>
Model instance filter	If using <i>auto-filter</i> , leave blank. If not using <i>auto-filter</i> , select your MIF for user filtering.
Include the standard user workflows	
Disabled Actions	Update <i>true</i> (checked). Note that you will need to uncheck <i>Remove</i> if using a MIF.
Purge Unmatched Records	Set to <i>true</i> (checked)

### Scheduling syncs

Consider the following with regards to scheduling syncs:

- MS365 user sync
  - Run and complete the MS365 user sync *before* running the Teams user sync. This ensures that any new or removed users in Entra are in the system and can drive the *auto-filter* behavior during the Teams user sync.
  - Run the MS365 user sync more frequently than the Teams sync to minimize delays for user onboarding.

The MS365 sync brings users into Automate and can then be run through the onboarding process (whether admin initiated or via flow through provisioning).

Typically, the Teams user is not required in the system to make onboarding possible. The onboarding workflow target syncs the Teams user in during the workflow.
- Teams user sync
  - May not need to be run as frequently as the MS365 sync.
  - If you're performing many user transactions outside the system (such as assigning numbers), the sync may need to be more frequent, for example, to ensure the accuracy of the number inventory.

Typically, updates won't need to be scheduled as the system updates data when records are opened. However, update syncs can be set up and scheduled as needed.

#### 8.6.2. Configure Microsoft tenant connection parameters

Microsoft

provider

**Note:** References in this section to “PowerShell Proxy” refer to the MS Windows host running PowerShell commands. References to just “Proxy” refer to the HTTP proxy server that the the Windows PowerShell host uses to access the tenant in the cloud.



This procedure configures the following connections:

- From Automate to the PowerShell Proxy
- Between the PowerShell Proxy and the tenant
- The Graph API connection between Automate and the tenant

The screenshot shows the 'Microsoft Tenant' configuration page in the Synergy application. The breadcrumb navigation at the top indicates 'Microsoft Tenant > Synergy'. The page is divided into several sections:

- Tenant:** Contains fields for 'Name' (filled with 'Synergy') and 'Description' (filled with 'New Tenant with All Licenses for').
- Microsoft Application Authentication:** Contains fields for 'Client ID' (filled with '029baa50-0c79-469b-9a96-bf647'), 'Tenant ID' (filled with '18b54a78-571c-4e5f-8df1-a1508f'), 'Certificate' (with a dropdown arrow), 'Certificate Thumbprint' (filled with 'ACB63C705F4F44EE02AA110792B1046Cf'), and 'Secret' (with a 'Show' checkbox).
- Powershell Server Authentication:** Contains fields for 'Host' (filled with '10.120.10.2'), 'Username' (filled with 'WSMan-svc'), and 'Password' (masked with asterisks, with a 'Show' checkbox).
- Microsoft Teams Admin Account:** Contains a field for 'Admin Username'.

### Prerequisites:

You will need:

- The FQDN or IP address of a single-node PowerShell Proxy, or the FQDN corresponding to your load balancer's virtual IP address. See:
  - Run PowerShell proxy server setup script in the Core Feature Guide
- The credentials for the local service account you created on the PowerShell Proxy
- Proxy authentication credentials (if the outbound Internet Proxy requires authentication)

---

**Note:** Authenticated proxy is not supported.

---

- The client ID and tenant ID
- Either the client secret (supported for Teams and Graph only, not Exchange) or the certificate (supported for Teams, Graph, and Exchange, and mandatory for Exchange), that you created when registering Automate as an application object with Microsoft Entra ID. For greater security, certificate is preferred.

If an Arbitrator is configured on Automate, the secret is required. See:

- Arbitrators in the Core Feature Guide

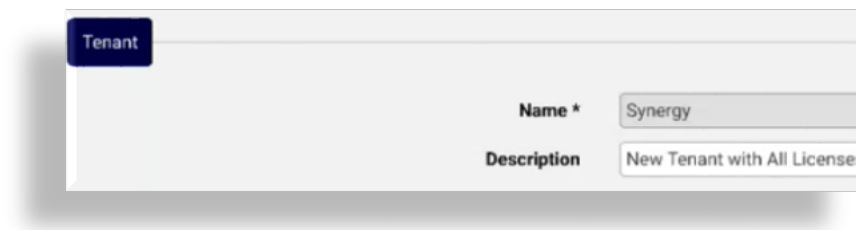
## To add and configure the Microsoft Tenant

**Tip:** Use the Action search to navigate Automate

1. Log in to the Automate Admin Portal as a Provider administrator, then go to the **Microsoft Tenant** page.

**Note:** By default, the Provider administrator role is the only role that has the ability to create Tenant connections.

2. Click the Plus icon (+), then choose the hierarchy, typically, Customer.
3. Fill out a name and a description for the tenant.



4. At **Microsoft Application Authentication**, fill out values for the application authentication:

**Note:** For more details around configuring application registration, see *Shared central application registration*:

- Fill out the **Client ID** and **Tenant ID** values recorded in the application registration setup.

**Note:** The tenant ID and the client ID identify the tenant in the Microsoft cloud.

- Select a certificate from the drop-down.

**Note:** It is strongly recommended that you use a certificate and not a secret. Secret is used only if you don't select a certificate. If you have a certificate and you select it here, the certificate is used for authentication for MS Graph, MS Teams, and MS Exchange.

If an Arbitrator is configured on Automate, the secret is required. See:

Arbitrators in the Core Feature Guide

If you're using MS Exchange, you *must* use a certificate as secrets are not supported for MS Exchange.

The value for **Certificate Thumbprint** is auto-populated when you select the certificate.

If you set up *shared central application registration* with the certificate you'll need to import the certificate into Automate then select it here when adding the tenant.

**Note:** For details, see:

### Shared central application registration in the Core Feature Guide

Choose an option, either of the following:

- Generate a certificate in Automate and upload it to MS Entra ID. See:
  - \* Generate a certificate for application registration in the Core Feature Guide
- If you already have a signed certificate from another source in your organization and it's already uploaded to MS Entra ID, you can upload that certificate into Automate and have Automate manage it on itself and on the PowerShell proxy. See:
  - \* Upload a certificate to use for app registration in the Core Feature Guide

- Fill out the **Secret** if an Arbitrator is configured on Automate, the secret is required. See:
  - Arbitrators in the Core Feature Guide

#### 5. Configure PowerShell server authentication details:

- At **Host**, fill out the FQDN or IP address of a single-node PowerShell proxy, or the FQDN corresponding to your load balancer's virtual IP address.

**Note:** For details around the local hosts file and the TrustedHosts WinRM configuration, see:

- Run PowerShell proxy server setup script in the Core Feature Guide

- At **Username** and **Password**, fill out the credentials for the local service account you created on the PowerShell Proxy. See:

- Run PowerShell proxy server setup script in the Core Feature Guide

#### 6. At **Microsoft Teams Admin Account**, ignore these fields unless you must use basic authentication (basic auth).

By default, the **Resource Account Basic Authentication** checkbox is clear (disabled), which means that you can only import (sync in) resource accounts. You'll need to enable this functionality to add, modify, or delete resource accounts with basic auth in Automate. However, once Microsoft enforces multifactor authentication, the ability to add, modify, or delete resource accounts in Automate will no longer be possible, regardless of whether this checkbox is selected.

**Important:** Basic auth requires service account credentials and will stop working when Microsoft enforces multi-factor authentication on all service accounts (starting July 2024). It is strongly recommended that you use application authentication instead of basic auth. See *Shared central application registration*.

If you must use basic auth, you'll need to contact system support for assistance to set up the required service account.

7. Configure outbound internet proxy connection parameters in the **PowerShell Server HTTP proxy** fields:

- If you have an outbound internet proxy deployed between the PowerShell proxy and the public internet, select **Use HTTP Proxy**.

**Note:**

- If there is no outbound Internet Proxy deployed between the PowerShell and the public internet, leave both **Use HTTP Proxy** and **Use HTTP Proxy Authentication** unchecked, and leave the **Username** and **Password** fields blank.
- Authenticated proxy is supported.

- If the outbound Internet proxy requires authentication, select **Use HTTP Proxy Authentication**, fill out a username and password.

**Note:** You will have already provisioned the outbound internet proxy's IP address (or FQDN) and port number when you set up the PowerShell proxy. See *Run PowerShell proxy server setup script*, and note the caveat regarding proxy authentication described at:

- PowerShell proxy deployment topologies in the Core Feature Guide

8. Add **Microsoft 365 HTTP Proxy** details to the Microsoft tenant:

- At **MS 365 HTTP proxy** / **MS 365 HTTPS proxy**, set the outbound internet proxy server if required for traffic outbound to the public internet.

The proxy setup defines the route for the MS Graph API communications that the system uses for communication with the MS 365 Cloud tenant.

- For HTTP proxy traffic, fill out a MS 365 HTTP proxy value with the following format: `http(s)://[user:password]@host:port/`. Special characters in either the user or password must be URL encoded. Verify the required format with the proxy administrator.
- For HTTPS proxy traffic, fill out a MS 365 HTTPS proxy value with the following format: `http(s)://[user:password]@host:port/`. Special characters in either the user or password must be URL encoded. Verify the required format with the proxy administrator.

---

**Note:**

- **MS 365 HTTP proxy** and **MS 365 HTTPS proxy** values will almost certainly be identical unless your proxy administrator has clearly told you that HTTP and HTTPS traffic are being proxied through different servers. It is not required that the **MS 365 HTTP proxy** address begin with `http://` or that the **MS 365 HTTPS proxy** address begin with `https://`. It is perfectly acceptable to proxy HTTP traffic to an `https://` address or HTTPS traffic to an `http://` address.
  - In both cases the host can be a FQDN if resolvable via DNS or the IP address of the internet proxy.
- 

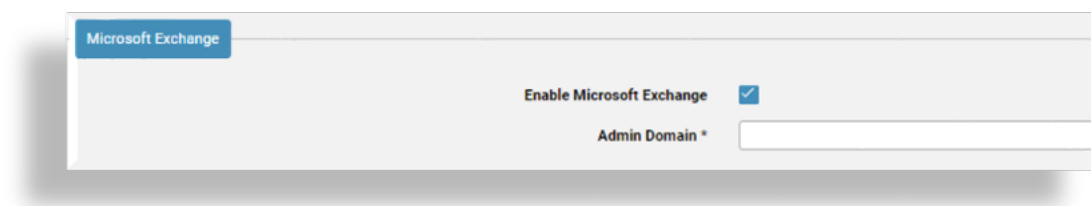
9. At **Microsoft Exchange**:

- If you're using Automate to manage Microsoft Exchange online, select **Enable Microsoft Exchange**.
- At **Admin Domain**, specify the Microsoft Exchange admin domain (the domain of the tenant) used to authenticate.

---

**Note:** The **Admin Domain** field displays once you select **Enable Microsoft Exchange**.

---



10. Configure advanced settings. The table describes fields in the **Advanced Settings** fieldset:

Field	Description
MS 365 MsolUser page size	Defines the maximum number of records to retrieve at a time from the API related to device/msgraph/MSOLUser. For optimal performance, leave this field blank to use the default value, 999.
MS 365 Group page size	Defines the maximum number of records to retrieve at a time from APIs related to device/msgraph/Group. For optimal performance, leave this field blank to use the default value, 999.
MS Teams Number page size	Defines the maximum number of records to retrieve at a time from APIs related to the msteamsonline/Number device model. Starting January 2025, Microsoft has set the page size limit for phone number retrieval to a maximum of 1000 numbers per query. Requests exceeding this limit will result in an error. See the Automate 24.2-PB1 release notes for the upgrade notes for EKB-22837, for details.
MS Teams CsOnlineUser page size	Defines page size for the msteamsonline/CsOnlineUser device model. Only set this value if the amount of data returned when using the default value is causing an error.
Maximum Rendered Template Size	The maximum allowed size of any rendered template for all models for all managed drivers. Size in bytes, so 900000000 is 900MB. Only set this value if the amount of data returned when using the default value is causing an error. Default is 900MB.
Auto filter Teams users	Defines whether to add a default, automatic filter to all CsOnlineUser syncs to only return records matching MsolUsers in the cache. Default is False. When enabled (True), no other sync filters can be used for CsOnlineUser syncs. Using additional filters will trigger a system error in this case.
Cloud environment type	The cloud environment type to authenticate to. Automate's default cloud environment for a Microsoft tenant is "Commercial". On upgrade or install, the cloud environment type is set to this default value. Automate also supports authentication to high security cloud environment types, allowing an admin to identify their Microsoft customer tenant as one that is operating in a high security cloud environment, either DoD (United States Department of Defence) or GCCH (Microsoft 365 Government Community Cloud High). When the tenant cloud environment type is set to either DoD or GCCH, Automate sets the appropriate environment names for the Microsoft Teams and Microsoft Exchange PowerShell modules, and appropriate URLs are used for the Microsoft Graph API.

**Note:** For details on the auto filter for Teams users, see:

Microsoft syncs in the Best Practices Guide

#### 11. Save your changes.

---

**Note:** When saving the tenant with the certificate selected, the certificate is deployed to the PowerShell proxy and installed. You can then import the certificate on the tenant App registration in Microsoft Entra for authentication of all apps (MS Exchange, Teams, and Graph).

---

12. Test your Microsoft tenant connection. You will be prompted to confirm the test.

---

**Note:** In this step you're verifying that Automate can connect to the Microsoft tenant using the Teams and Exchange Powershell modules on the Windows server as well as using the Graph API on the Automate platform.

---

- On the **Microsoft Tenant** page, choose the relevant tenant.
- Click **Test Connection**.

### Modifying a Microsoft Tenant

Modifying a Microsoft tenant will only overwrite those driver parameters in the underlying connections (MSTeamsOnline, MSExchangeOnline, MSGraph) that are managed by the tenant workflows. Other driver parameters will be left as is.

Also refer to the **Advanced Settings** above to modify the *page size* options for the Microsoft Tenant in order to adjust the synchronization performance.

### Next steps

- Verify that no changes are needed in user name mapping macros prior to sync. High level administrators with access to the data/MultivendorUsernameMappingMacros model instances should carry out this task.

---

**Important:** For release 21.5-PB5, Multivendor environments using the data/MultivendorUsernameMappingMacros model at a hierarchy *below* sys level require an additional update. High level administrators with access to this model should ensure instances include:

```
"username_macro_ms_365": [
  "{{ input.UserPrincipalName }}",
  "(( fn.is_none_or_empty input.username == fn.true ))<NOT_FOUND>(( data.User.username_
↪ | username:input.username != ' ' ))<{{ data.User.username | username:input.username }}>
↪<NOT_FOUND>",
  "(( fn.is_none_or_empty input.username == fn.true ))<NOT_FOUND>(( data.User.username_
↪ | username_ms_365:input.username != ' ' ))<{{ data.User.username | username_ms_365:input.
↪ username }}><NOT_FOUND>",
  "(( fn.is_none_or_empty input.UserPrincipalName == fn.true ))<NOT_FOUND>(( data.User.
↪ username | email:input.UserPrincipalName != ' ' ))<{{ data.User.username | email:input.
↪ UserPrincipalName }}><NOT_FOUND>",
  "(( fn.is_none_or_empty input.UserPrincipalName == fn.true ))<NOT_FOUND>(( data.User.
↪ username | username_ms_365:input.UserPrincipalName != ' ' ))<{{ data.User.username |
↪ username_ms_365:input.UserPrincipalName }}><NOT_FOUND>",
  "(( fn.is_none_or_empty previous.UserPrincipalName == fn.true ))<NOT_FOUND>(( data.
↪ User.username | username_ms_365:previous.UserPrincipalName != ' ' ))<{{ data.User.
↪ username | username_ms_365:previous.UserPrincipalName }}><NOT_FOUND>",
```

(continues on next page)

(continued from previous page)

```

    "(( fn.is_none_or_empty input.UserPrincipalName == fn.true ))<NOT_FOUND>(( data.User.
↪username | username_ms_teams:input.UserPrincipalName != ' ' ))<{{ data.User.username |_
↪username_ms_teams:input.UserPrincipalName }}><NOT_FOUND>",
    "(( fn.is_none_or_empty previous.UserPrincipalName == fn.true ))<NOT_FOUND>(( data.
↪User.username | username_ms_teams:previous.UserPrincipalName != ' ' ))<{{ data.User.
↪username | username_ms_teams:previous.UserPrincipalName }}><NOT_FOUND>",
    "(( fn.is_none_or_empty input.UserPrincipalName == fn.true ))<NOT_FOUND>(( device.cucm.
↪User.userid | userIdentity:input.UserPrincipalName != ' ' ))<{{ device.cucm.User.userid_
↪| userIdentity:input.UserPrincipalName }}><NOT_FOUND>",
    "(( fn.is_none_or_empty input.UserPrincipalName == fn.true ))<NOT_FOUND>(( device.cucm.
↪User.userid | mailid:input.UserPrincipalName != ' ' ))<{{ device.cucm.User.userid |_
↪mailid:input.UserPrincipalName }}><NOT_FOUND>"
  ],

```

- Perform a sync from the Microsoft tenant to import Microsoft users, tenant dial plan, licenses, and policies to the customer level. You will be prompted to confirm the syncs.

For Microsoft Exchange, ensure that instances for all 4 device models (User mailboxes, Shared Mailboxes, Room Mailboxes, and Distribution Mailboxes) are synced in at the level where the tenant exists.

- Configure the customer-wide site defaults doc (SDD), CUSTOMER\_TEMPLATE. See *Site Defaults Doc templates*.
- Add network device lists (NDLs) with Microsoft 365 and Microsoft Teams tenant details. NDLs are required when adding sites. See:
  - Network Device Lists (NDLs) in the Core Feature Guide
- Create sites.
- Run the overbuild. See:
  - Overbuild for Microsoft in the Core Feature Guide
- Go to:
  - Configure Automate for Microsoft services in the Core Feature Guide

## Related topics

- Microsoft Quick Start Guide for Automate in the Core Feature Guide
- Microsoft Overview in the Core Feature Guide
- Microsoft syncs in the Best Practices Guide
- Site Defaults Doc templates in the Core Feature Guide
- Run PowerShell proxy server setup script in the Core Feature Guide



## 9. Data collection

### 9.1. Recommended RIS API data collector interval

It is recommended that, to determine the interval that the (RIS) data collector service should poll the Cisco UCM, consider that it takes approximately 14 minutes to collect information for around two hundred thousand phones on a cluster.

The default value of the **RIS API data collector interval**, 43200 seconds (12 hours), can be adjusted accordingly.

---

**Note:** Collection processes run in parallel for each UCM on Automate.

---

To adjust the value, refer to the System Monitoring Configuration section in the Advanced Configuration Guide.

## 10. API performance

### 10.1. API resource listing best practice

This section provides best practices when using API GET requests when listing resources, and examines best practices for using a number of API request parameters and parameter values.

**Note:** For further details on API parameters, refer to the API Guide.

The table describes API request parameters for resource listing:

Parameter	Description	Value	Default
skip	The list resource offset as a number.		0
limit	The maximum number of resources returned. The maximum value is 2000. If the Range request header is used, it will override this parameter.	1-2000	50
count	Specify if the number of resources should be counted. If false, the pagination object in the response shows the total as 0, so no total is calculated and the API performance is improved.	true, false	true
order_by	The summary attribute field to sort on.		First summary attribute
direction	The direction of the summary attribute field sort (asc:ascending, desc: descending).	asc, desc	asc
summary	Only summary data is returned in the data object.	true, false	true
policy_name	Return a model form schema where the Field Display Policy with name [FDP name] is applied to it. Use policy with the parameters schema and format=json.	[FDP name]	
cached	System will respond with resource information where the data was obtained from cache. (Functionally only applicable to device models and data models).	true, false	true

The table lists parameters with their best practice recommendations and considerations:

coun	The value of <code>count=true</code> is very expensive in terms of performance, and more so as the size of the resource grows. The first count query of, for example, a 36 000 Data Number Inventory resource can take as long as a minute to return a response. However, subsequent calls should decrease in execution time. The value <code>count=true</code> should only be used if it is unavoidable. An alternative is to iterate over pages ( <code>limit=200</code> ) until the request returns less than 200 instances, or to simply paginate until no more resources are returned.
orde	No performance change if another summary attribute is specified.
dire	No performance change if either values <code>asc</code> or <code>desc</code> are used.
poli	The parameter is used by the GUI for display purposes. Timing data shows that the initial call with this parameter takes longer than subsequent ones, possibly because of cache priming after a restart. Subsequent calls show the execution time is on par with requests that do not include the parameter.
summ	Depending on the data required by the request, time can be saved if the value <code>summary=true</code> , so that only the summary data is returned.
limi	Execution time and memory consumption is impacted if the <code>limit</code> value is large.

The recommended parameter values for an optimal API list request (GET) are as follows:

- `cached=true`
- `summary=true`
- `count=false`
- `policy_name` not used

Example results with various parameter values (36 000 Data Number Inventory resource):

```
count:true, skip:0, policy_name:, limit:200, summary:false in 6.51744103432 s
count:true, skip:0, policy_name:, limit:200, summary:true in 5.6118888855 s
count:false, skip:0, policy_name:, limit:200, summary:false in 1.55350899696 s
count:false, skip:0, policy_name:policy_name=HcsDNInventoryDatFDP, limit:200,
summary:true in 5.17663216591 s
count:false, skip:0, policy_name:, limit:200, summary:true in 1.09510588646 s
```

## 10.2. Long running API requests

### 10.2.1. Overview

To optimize memory utilization and performance, the system is configured so that the API server will manage workers with the following defaults:

- After receiving a restart signal, workers have 100 minutes to finish serving requests
- A random restart interval of between 0 and 600 requests per worker (4 workers per node, 4 nodes in a cluster)

The recommended API best practice is to schedule and then poll transactions, since long running requests can affect recycling; that is, preferably short requests and then poll.

### 10.2.2. Polling example

To retrieve the status of a given transaction:

```
GET /api/tool/Transaction/[pkid]/poll/?format=json
```

The response contains essential status of the transaction, for example:

```
{
  [pkid]: {
    status: "Success",
    href: "/api/tool/Transaction/[pkid]",
    description: "Name:RDP-auser1857 Description:RD for auser1857"
  }
}
```

---

**Note:** Refer to *Poll Transactions* and *Example of an Asynchronous Mutator Transaction with nowait=true* in the API Guide.

---

# 11. System maintenance

## 11.1. Transaction archiving

The following are considerations when determining the frequency of the transaction archiving schedule to set up on the system. If a schedule is not set up for transaction archiving, system Alerts will be raised as well as a warning on the platform CLI login:

TRANSACTION DATABASE MAINTENANCE NOT SCHEDULED

- Run `voss transaction count <days>` on your system to inspect the number of transactions during a given period to determine your usage metrics.

Refer to the *Database Commands for Transaction Management* topic in the Platform Guide for details on transaction archive command use and scheduling:

- **voss transaction delete <days>**
- **voss transaction export <days>**
- **voss transaction archive <days>**

- Business policies - company policies may drive your choices: the immediate access to transaction logs for a period of time, security policy on data/audit retention, and so on.

---

**Note:** The transaction archive process does mean the logs are not lost, just that they are not immediately accessible in the administrator graphical interface for searching.

---

- You can also set up system monitoring thresholds so that you receive alerts via the GUI and SNMP if the threshold is exceeded - which might indicate you need to review the archive schedule to increase how frequently it runs.

See the *SNMP* and *Automate System Monitoring Traps* topics in the Platform Guide.

## 11.2. Automated database cache cleanup

From Automate release 19.3.2 onwards, it is now longer necessary to schedule or manually manage the database cache optimization using the `voss trim-cache` command.

From release 20.1.1, this command is no longer available. A resource history is now maintained as a series of resource differences and is automatically optimized.

---

**Note:** A minimum retention period of seven (7) days is applied to resource differences in the resource history.

---

## 12. Admin Portal setup

### 12.1. Navigation - menu and dashboards

#### 12.1.1. Overview

Automate provides several tools for customizing the Portal experience to your requirements.

The Admin Portal uses two key ways to provide users with the means to navigate around the system to key features:

- Configurable navigation menus (on the left of the screen)
- Configurable Home page - this is the landing page you see when logging on or when clicking the Home button

#### Related topics

- Role-based dashboards and menus in the Core Feature Guide
- Dashboards in the Core Feature Guide
- Menu layouts in the Core Feature Guide
- Use the Action search to navigate Automate in the Core Feature Guide

#### 12.1.2. Navigation configuration options

The table describes configuration options to enhance navigation:

Configuration	Description
Naming conventions for menus and dashboard	Use naming conventions relevant to your users and organization. For example, use business-relevant names for admins, and technical terms only for advanced users.
Linking from menus	Links from menus are typically set up to the form/view or a list or to other system models that users need access to for various tasks.
Display policy	The display policy for views, or for when users choose an entity from a list view, which determines the form layout displayed to a user.
Configuration template (CFT)	If you've chosen a CFT, it is applied when a view is accessed from the menu item, or if the user clicks Add from a list view. The CFT can also define default fields (or read-only default values for read-only fields), as well as default values for fields that the CFT hides from view.
Fixed filters	Configured for a menu layout, and applied by default in the back-end and thus not visible to a user.
Configurable filters	May be fully or partially defined for menu layouts pointing to lists. This filter provides an interim step between clicking a menu item or link, and opening a list view. A configurable filter launches a pop-up allowing a user to enter filter criteria relevant to the menu item. When the user chooses the criteria the list view displays based on the criteria. Users can view, modify, or delete configurable filters.

**Note:** See the Core Feature Guide for more information on navigation configuration options.

### 12.1.3. Navigation strategies

It's important that you provide efficient methods for users to navigate through the system and to access required functionality.

Ensure frequently used functionality is easy to reach for different types of users. Create menus and the dashboard based on the requirements of different user roles, and review these requirements regularly to ensure the greatest efficiency and user experience.

The table outlines key efficiency outcomes for enhancing navigation and the user experience:



Goal	Description
Quick Access to tasks and searches	<p>Populate dashboards with frequently used tasks and searches to provide access with one click, and the ability to easily return to quick access functionality via a Home icon.</p> <p>EXAMPLES:</p> <ul style="list-style-type: none"><li>• Access to top MACDs on dashboards, with appropriate FDPs and CFTs. See Feature Experience.</li><li>• Add saved searches with predefined filter criteria as links. For example, a saved search for the list of unregistered phones, linking to phones with criteria set to <i>status starts with unregistered</i>. Note that you'll only have access to the Saved Searches functionality if your access profile permissions allow it (read permissions on data/UserSavedSearch).</li><li>• Links with configurable search criteria, for example, to find a phone by user. This can be done with a link for Phone, and filter criteria set to ownerID. In this case, the user is prompted to provide the user name to find a phone.</li></ul> <p>If you don't have enough space on a dashboard for all the frequently accessed tasks you wish to add, add these items to the menus.</p>

Goal	Description
Feature experience	<p>Instead of creating one link to a feature that has many scenarios, you may want to include multiple menus or dashboard entries to the same feature, but use different display policies, CFTs, and filter options for specific use cases.</p> <p>This can be a very simple way to create an experience of a feature for specific scenarios, and prevent relying on users to follow a procedure or enter specific information.</p> <p>EXAMPLES:</p> <ul style="list-style-type: none"><li>• Create a link to add SIP trunks as part of a regularly added 3rd party application integration. This can link to the SIP trunk device model, and use a display policy that shows only settings requiring user input (such as IP address and the port for the remote system). The CFT could predefine all other technical settings based on the scenario (for example, CSS, call presentation details, digit manipulation, and so on).</li><li>• Create lines for different scenarios. Use display policies to show only those fields that must have user input, while CFTs pre-populate settings based on the scenario. You can add menu items for line type A, or line type B, for example, to cover the various scenarios, and combine this with filters in the menus and landing pages to differentiate between line types in the list views.</li><li>• UCM feature management: UCM can be managed in Automate by accessing the device models directly. The API definition from Cisco drives the default device model layout, and may include field names and ordering that doesn't align with the Automate Admin Portal experience. You can improve this display policies to impose your preferred order and field labels. Combine this with CFTs to predefine default values or to simplify user input requirements to reduce possible setup errors.</li></ul>

# Index

## V

### voss

- voss queues, [26](#)
- voss transaction archive, [73](#)
- voss transaction count, [73](#)
- voss transaction delete, [73](#)
- voss transaction export, [73](#)
- voss workers, [9](#), [18](#)

## W

### web

- web service, [9](#), [18](#)