
Whitepaper

CSA-UCMDB integration

Contents

Introduction	3
Configuring CSA for Integration with UNIVERSAL CMDB	3
UNIVERSAL CMDB Provider Configuration	4
CSA Component Types aligned with UNIVERSAL CMDB's CI Types	5
Creating a Custom Component Type in CSA for a UCMDB CI Type	5
Designer Visible	7
UNIVERSAL CMDB CI Attribute Mapping to CSA Component Property	8
UNIVERSAL CMDB CI Attribute Type	8
Deploying a sequenced design to demonstrate UCMDB integration	9
CI Instance Id	10
Lifecycle Phases where the CIs are created and updated	11
Component properties as CIs	11
Additional CIs	11
Modify Subscription and Service Instance Upgrade	12
Group Owned subscriptions	12
Multitenancy with UCMDB	12
FAQ	13
Appendix	14
OOTB Components Hierarchy based on CI Types	15
CI Types	16
Reconciliation in UCMDB	19
Send documentation feedback	20
Legal notices	20

Introduction

The Universal CMDB, a configuration management database solution, automatically collects and manages accurate and updated business service definitions, associated infrastructure relationships and detailed information of assets.

It is a central component in many of the key processes in your IT organization, such as change management, asset management, service management and business service management. UNIVERSAL CMDB ensures that these processes rely on comprehensive and true data for all business services. Together with UNIVERSAL CMDB Configuration Manager (UCMDB CM) you can standardize your IT environments, and make sure they comply with clear policies and defined authorization processes.

Integration of Cloud Service Automation (CSA) and UNIVERSAL CMDB automates the creation and maintenance of configuration items for resources provisioned in a service by CSA, along with relationship between configuration items which can be leveraged by management activities by other management tools or use cases e.g. event correlation, universal discovery etc.

CSA (CI Type based) is a new component palette available for Sequenced Designs. The component types in the CSA (CI Types based) palette are aligned with UNIVERSAL CMDB's CI Type system, so that each new CSA component type is mapped to its corresponding CI Type in UNIVERSAL CMDB, and the attributes of the corresponding CI Types are mapped to the properties of the CSA component type.

When a Service Design is built using components based on this new component type system, and UNIVERSAL CMDB provider is configured, then the equivalent service topology will be created in UNIVERSAL CMDB as the service design is deployed, or a service offering created such service design is subscribed in CSA.

Configuring CSA for Integration with UNIVERSAL CMDB

Following are the major steps to integrate CSA with UCMDB:

1. [The UNIVERSAL CMDB Provider should be configured in CSA.](#)
2. [Create a Custom Component in CSA for a UCMDB CI Type.](#)

Use an existing component that is based on CI Type based component type, or create a custom component type from the new component types.

3. [Create the sequenced design and publish its corresponding offering to the relevant catalog.](#)

When you successfully create a subscription in CSA, you will see that the CIs and deployment topology would be created in UCMDB.

UNIVERSAL CMDB Provider Configuration

An UNIVERSAL CMDB provider must be configured in CSA for a successful integration. Complete the following steps to configure UNIVERSAL CMDB as provider in CSA:

1. Login to the server where UNIVERSAL CMDB is installed and extract the UCMDB server certificate using the following command:

Note: The UCMDB certificate must be imported into CSA's keystore if the UCMDB provider is configured to use secure HTTP for communication.

For Windows:

```
%UCMDB_JRE_HOME%\bin\jre\bin\keytool.exe -exportcert -keystore  
%UCMDB_JRE_HOME%\conf\security\server.keystore -alias hpcert -storepass hppass -file  
remoteServer.cert
```

Where, %UCMDB_JRE_HOME% is the directory where the Java Runtime Environment used by UNIVERSAL CMDB is installed.

For Linux:

```
$UCMDB_JRE_HOME/bin/jre/bin/keytool -exportcert -keystore  
$UCMDB_JRE_HOME/conf/security/server.keystore -alias hpcert -storepass hppass -file  
remoteServer.cert
```

Where, \$UCMDB_JRE_HOME is the directory where the Java Runtime Environment used by UNIVERSAL CMDB is installed. Copy the file "remoteServer.cert" to the server where CSA is installed, to a temporary location.

2. Open a Command Shell server where CSA is installed, change to the following directory and execute the command

For Windows:

Change to,

```
%CSA_HOME%\openjre\bin keytool.exe -importcert -alias hpcert -file <FILE  
PATH>/remoteServer.cert -keystore "%CSA_HOME%\openjre\lib\security\cacerts
```

For Linux:

Change to,

```
$CSA_HOME/openjre/bin keytool -importcert -alias hpcert -file <FILE  
PATH>/remoteServer.cert -keystore "$CSA_HOME/openjre/lib/security/cacerts
```

Where, <FILE PATH> is the path where the "remoteServer.cert" file is copied to as in step 2.

Note: If CSA is using an external JRE, then import the UNIVERSAL CMDB certificate into the keystore of the external JRE.

3. Restart CSA services after importing the certificate.

Once these steps are executed, create a new UNIVERSAL CMDB provider in CSA. If a UNIVERSAL CMDB provider is already configured, ensure the certificate is exported from the configured UNIVERSAL CMDB instance and imported into CSA.

Notes:

- When designs having CI Type components are subscribed; during service provisioning if UCMDB is not reachable, the subscriptions will not fail. Subscriptions will be active however, the topology in UCMDB will not be created.
- Only one UCMDB provider is supported.

CSA Component Types aligned with UNIVERSAL CMDB's CI Types

The new component types are included in a palette called **CSA (CI Types based)**. The component types in this palette are equivalent to similarly-named CI Types in the UCMDB type system. The new component types cover most commonly used types in Compute, Network, Storage and Software.

To see the complete list of the out-of-box CI Type Components in UNIVERSAL CMDB palette, [click here](#).

Creating a Custom Component Type in CSA for a UCMDB CI Type

An UNIVERSAL CMDB provider should be configured if the new component types based on UCMDB's CI Type system needs to be created in UCMDB, or if the service topology provisioned by CSA should be created in UCMDB.

This section describes the steps and prerequisites for creating a custom CSA component type. This is required if a component type equivalent to a CI type does not exist in CSA.

Prerequisites

The following are the prerequisites for defining the CI Type for a new component type:

1. UNIVERSAL CMDB provider must be configured. See, [Configuring CSA for Integration with UNIVERSAL CMDB](#) section for details.
2. The CI Type must exist in the configured UCMDB server.
3. The CI Type of the base component type must be the parent of the CI Type set for the new component type, or it must be the same as the CI Type set for the new component type.

Note: The input for CI Type is not mandatory and can be left blank. When left blank, the CI Type of the base component type is used for the new component type.

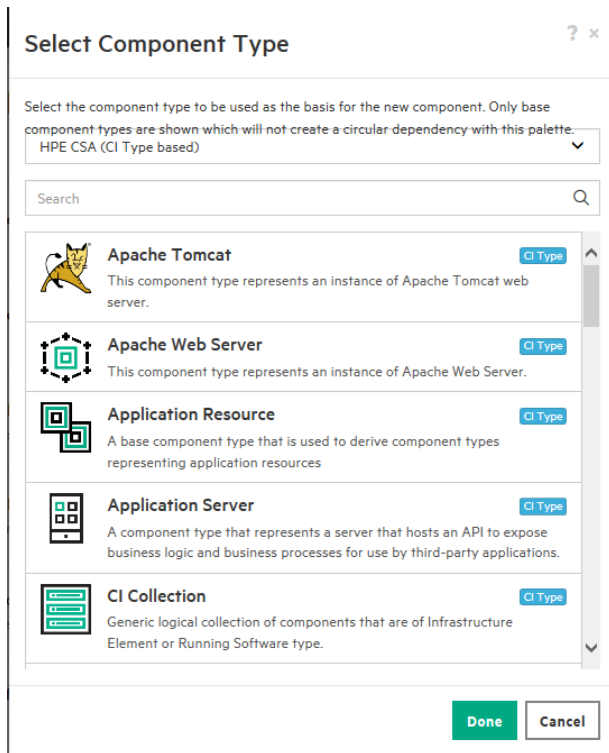
4. The UCMDB server certificate must be imported into CSA's keystore if the UCMDB provider is configured to use secure HTTP for communication.

Follow the steps below to create a custom CSA component type:

1. From the **Cloud Service Automation Dashboard**, select **Designs > Sequenced > Components**. The **Sequenced Components** page is displayed.
2. CSA (CI Type) based palette from the left panel > click the gear icon and select **Create Component Type**. The **Create Component Type** dialog box is displayed.
3. Select a CSA component type as the Base Component Type.

Base Components that are mapped to CI Types in UNIVERSAL CMDB is indicated by the **CI Type** label displayed beside component type, as shown in the image below.

Component Types indicated with the Label CI Type



Points to Note:

In CSA (CI Type based) Palette

- You can only create CI Type based components. Although other palettes can be selected from the Select Component Type dialog box, components for those palettes will not be displayed. Only components of CSA (CI Type based) will be displayed.
- The **CI Type** field is visible in the **Create Component Type** dialog box by default.
- Selecting a Base Component is mandatory.
- When a Base Component is selected, the internal name of the selected Base Component is displayed in the CI Type field by default. However, you can change the CI Type to any other related CI Type component. But if the CI Type is not related, then an error is shown and component type will not be created.
- When the CI Type field is left blank, the selected Base Component is considered as the CI Type.

In palette other than CSA (CI Type based)

- You can create components of all types including CI Type by selecting the Base Component of CI Type.
- The CI Type field in the Create Component Type dialog box is not visible by default. Only on selecting a CI Type component as the Base Component, the CI Type field is visible.
- Selecting Base Component is not mandatory for components other than CI Type.

4. Enter a name for the CI Type.

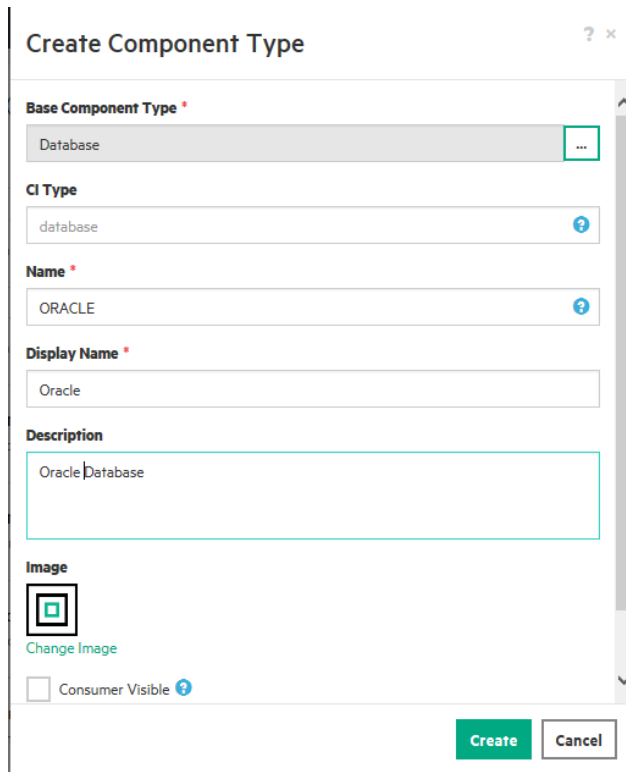
The new component type should be mapped to in UNIVERSAL CMDB. This CI Type must exist in UNIVERSAL CMDB else, the component type creation will fail.

5. Enter a name and display name for the new component type.

Note: You must also have a valid UNIVERSAL CMDB provider defined in CSA as described in the section [Configuring CSA for Integration with UNIVERSAL CMDB](#).

6. Select the **Designer Visible** check box to set the component type to be visible and available for use in service designs.
7. Select the Consumer Visible check box to set the component type to be visible in the Marketplace Portal in Marketplace Portal.
8. Select the Pattern check box to set the component of this type to be accompanied by a Clone Pattern action in the parent service component.

The image below shows how a new CI Type component can be defined in CSA.



The screenshot shows a 'Create Component Type' dialog box. It has a title bar with a question mark and a close button. The main content area is divided into several sections: 'Base Component Type' with a dropdown menu showing 'Database'; 'CI Type' with a text input field containing 'database'; 'Name' with a text input field containing 'ORACLE'; 'Display Name' with a text input field containing 'Oracle'; 'Description' with a text area containing 'Oracle Database'; 'Image' with a small square icon and a 'Change Image' link; and 'Consumer Visible' with an unchecked checkbox. At the bottom right, there are 'Create' and 'Cancel' buttons.

Designer Visible

When the Designer Visible check box for a component type is selected, then that component type is concrete and can be used in service designs. Such component types are visible in the Service Designer during design creation. Component types that are not Designer Visible can be used as base component types and new component types that are concrete in nature can be created and used.

For example, the Server Group and Server component types are marked as Designer Visible. This means a designer will be able to see the components of this type in the service designer, and can use them as components in a sequence design.

Component types such as, Node, Node Element and Running Software are not visible in the service designer. Instead it is recommended to use more concrete sub-types of these component types such as, Server, Network Connection or Application Server. However, you can create custom component types using such component types as base component types only, and use the custom component types while creating sequence designs.

Note: Components created using abstract components as base components cannot be set as designer visible. By default, the **Designer Visible** check box will be disabled on selecting an abstract component as the base component.

UNIVERSAL CMDB CI Attribute Mapping to CSA Component Property

A CI Type have many attributes of which there are key and mandatory to identify that CI uniquely and also participates in reconciliation of the CIs. When the new custom component is created based on a CI Type, the key properties needs to be created on the Component. The Property name on the component should match to the CI Attribute name in UCMDDB. If the key properties are not created or they are not set with the relevant information, the CI creation in the UCMDDB will fail during the fulfillment of the design using such components.

Note: The subscription will not fail if the CI creation is failed in UCMDDB.

For example, if you create a Component “Rack” for the CI “rack” from the UCMDDB, the key properties are name and location attributes such as row_name, space_name, grid_location, floor; then all these property needs to be created on the component Rack in CSA.

The additional properties can also be created on the component based on the CI Type attributes with the same name.

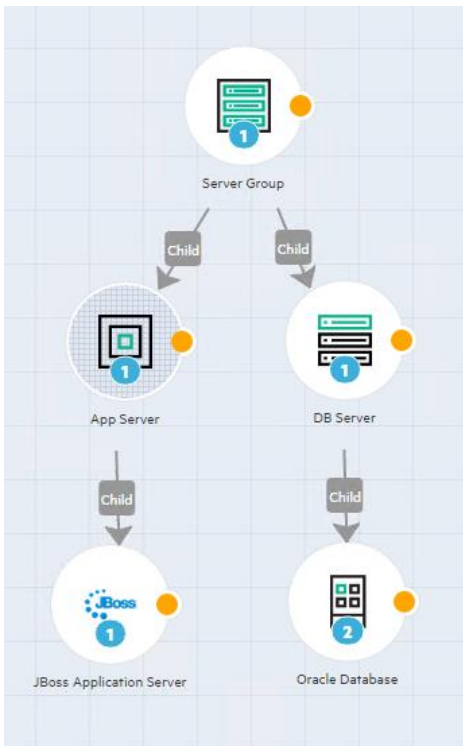
UNIVERSAL CMDB CI Attribute Type

UCMDDB Attribute Type	Supported (Yes/No)	CSA Property Type
String	Yes	String
Bytes	Yes	String – The value should be set as String type and it will be converted to bytes and stored in the UCMDDB
Boolean	Yes	Boolean
Integer	Yes	Integer
Long	Yes	String – The value should be set in the String type and it will be parsed back to Long data type and stored in UCMDDB.
Float	Yes	String – The value will be set in the String format and it will be parsed back to Float data type and stored in UCMDDB
Double	Yes	String – The value will be set in the String format and it will be parsed back to Double data type and stored in UCMDDB
Date	Yes	String - The value will be set in the String format which MUST be in the ISO 8610 specified format. And it will be parsed back to Date data type and stored in UCMDDB
List of Integers	No	The value is ignored
List of Strings	No	The value is ignored
XML	No	The value is ignored

Deploying a sequenced design to demonstrate UCMDB integration

This section describes creating a two tier sequenced design and provisioning. Once the provisioning is complete, you can examine the UCMDB to analyze the CIs, relations and the topology created.

1. Create a two tier sequenced design in CSA. The two tier design can include a Server Group, Server and Platform, typically it can be Application Server stack and Database stack. Add the required Resource Offering in order to realize the Server and Platform components. It can deploy on one of the IaaS providers like vCenter, Helion OpenStack etc., and use the PaaS providers like DMA.
2. Set the Clone Pattern actions and mark the Patterns for the components. Set the Subscriber options so that the stacks can be scaled out/in.
3. Publish the Design.
4. The following show a two tier sequenced design.



1. Create the service offering and publish it to relevant catalog.
2. Create a subscription
3. Once the subscription is online, check the UCMDB to analyze the CIs, relations and the topology which get automatically created in UCMDB.

Note: There can be additional CIs which are discovered from other sources contributing to the same topology and it will be shown under the same topology view.

CI Instance Id

During the Service Design deployment (fulfillment), if the design is using the CI Type based component and the UCMDB provider is configured then the equivalent CIs are created in the UCMDB and its relationship is updated accordingly. This will form a service topology in UCMDB based on the CSA deployment model and this topology can be used for all UCMDB use cases.

CI Instance Id is the property which is available in all the CI Type based components. This property will have the UCMDB CI Global ID which is obtained when the CI is created in UCMDB. The CI Instance ID of the component is updated in CSA before the Pre-Initialize phase of that component. The CIs attributes are updated at the end of each major lifecycle phase of the component as and when the component properties are updated. Once the service design is un-deployed and components are marked to be finalized, then the CIs are deleted from UCMDB.

The CI Instance Id can be used by other integrations (like SM) to refer to the CIs created by CSA and update them accordingly.

Lifecycle Phases where the CIs are created and updated

The CIs are created in the UCMDB during the fulfillment of the service component. The CIs are updated during multiple phases of the components lifecycle. The following are the lifecycle phases.

- **Before Pre-Initialize**

The initial CI is created with the key properties filled with the UUIDs of the component before the Pre-Initialize phase processing. The CI instance Id is updated on the component after the CI is created, so that the CI can be further accessed by any Actions using the CI Instance Id.

- **Post-Deploy, Modify**

The CI attributes are updated with the component properties.

- **Post-Undeploy**

The CIs are deleted when the components are finalized.

Note: From CSA 4.7, the following terms from provisioning stages in component lifecycle are renamed as given below:

Prior to CSA 4.7	With effect from CSA 4.7
Pre-Initialize	Before Initializing
Post-Deploy	After Deploying
Post-Undeploy	After Undeploying

Component properties as CIs

The component can have special properties which are CIs in the UCMDB. Currently, Server component has IP Address and MAC Address properties for which the CIs are created in the UCMDB when the Computer CI is created.

Additional CIs

In addition to the CIs created based on the deployment model from the sequenced designs, there are few more Business Service and other CIs which are also created. This helps in managing the topology and get the impact analysis for a CSA artifact. The following are the CIs which are created;

BusinessApplication

1. This is mapped to the Service Instance in CSA. For each Service Instance in CSA there will be Business Application CI created. The name is set with the UUID of the service instance.

Organization

2. This is mapped to the Organization in CSA to which the Subscription/Deployment belongs to. The organization is created in the UCMDB for the first time if it does not exist.

Person

3. This is mapped to the User who has created the subscription/deployment.

FunctionalGroup

4. This is mapped to the group in case the subscription is owned by a group.

CloudSubscriptionService

5. This is mapped to the Subscription in the CSA. For every subscription created in CSA there will be CloudSubscriptionService CI created. This CI has additional properties like start_date, end_date and subscription_status.

Note: This CI is not available OOTB with UCMDB. There is a new package available in ITOM Marketplace which should be deployed on the UCMDB. If there no such CI Type exists, then CSA will not create any CI instance.

To download the new package, go to **ITOM Marketplace > Cloud Service Automation > Resource tab > Tools** tile and download the **CloudSubscriptionService UCMDB Package**. Deploy the package using the UCMDB **Package Manager** from the **Administration** tab.

Modify Subscription and Service Instance Upgrade

During the subscription modification, the CIs are updated with the component properties to reflect the changes in the component properties which might have happened during the subscription modification. For instance, flexing memory might have increased the memory size and new memory size gets updated to the Computer CI.

The CIs are created for the new components which are created as part of flex out and CIs are deleted for components deleted as part of flex in during the subscription modification.

During the Service Instance upgrade, if new CI based components are created the same will be reflected in UCMDB.

Group Owned subscriptions

With the Business Application CI, the Person CI is created to represent the user who deployed the Service Instance. For the group owned subscriptions the FunctionalGroup CI is created to represent the Group which he subscription is owned by.

Multitenancy with UCMDB

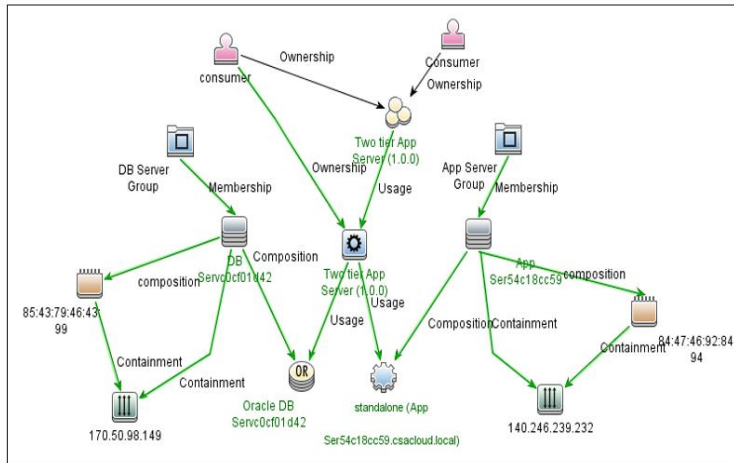
While deploying UCMDB, the multitenancy option should be enabled. If enabled then the OwnerTenant attribute of all the CIs will be updated with the respective Tenant ID. This will help in querying the CI based on the tenant information.

For Example

Design Name: Two tier App Server
Subscription: Two tier App Server
Organization: Consumer
User: Consumer



Two tier App Server Design in CSA



UCMDB Topology

FAQ

- **Is UCMDB integration is available for topology designs?**

No, the CSA platform level integration with UCMDB is available for only sequenced designs.

- **Is it possible to use existing component types with UCMDB?**

No, it is not possible to use existing component types with UCMDB. Only the new component types in the CI Type palette, or types that are derived from these new component types can be used with UCMDB.

- **Can the component types in the new palette be used with UCMDB prior to 10.3x?**

The component types in the new palette were tested with UCMDB 10.30. If there is a backward compatibility with older UCMDB releases, it may be possible to use the component types with them. But they were not tested with any release prior to 10.30.

- **Is it possible to use the new component types with a CMDB implementation other than UNIVERSAL CMDB?**

No, the CSA platform level integration is available only with UNIVERSAL CMDB.

- **The components in the new palette do not represent all CI types. Is it possible to create new types that are available in UCMDB but not in CSA?**

Yes, it is possible to create new component types for CI Types that do not exist in the new palette but are available in UCMDB.

- **Some of the components in the new palette do not contain corresponding properties for all the attributes in CI Types. Is it possible to create properties for missing attributes?**

It is possible to create properties for attributes that are available in UCMDB, but are missing from the component types in the new palette. However, it is advised to create a new component type using the existing type as a base component type and then add missing attributes as properties to the new component type.

- **When a service is cancelled, can the CI Instance ID be retained? Is it possible to configure the behavior so that some CI Instance IDs can be retained even after the services are cancelled?**

In CSA 4.8, it is not possible to retain the CI Instance ID when a service is cancelled. However, the requested enhancement is being planned in a future release and will be based on UCMDB's ageing policy for CI Instance IDs of CI Types.

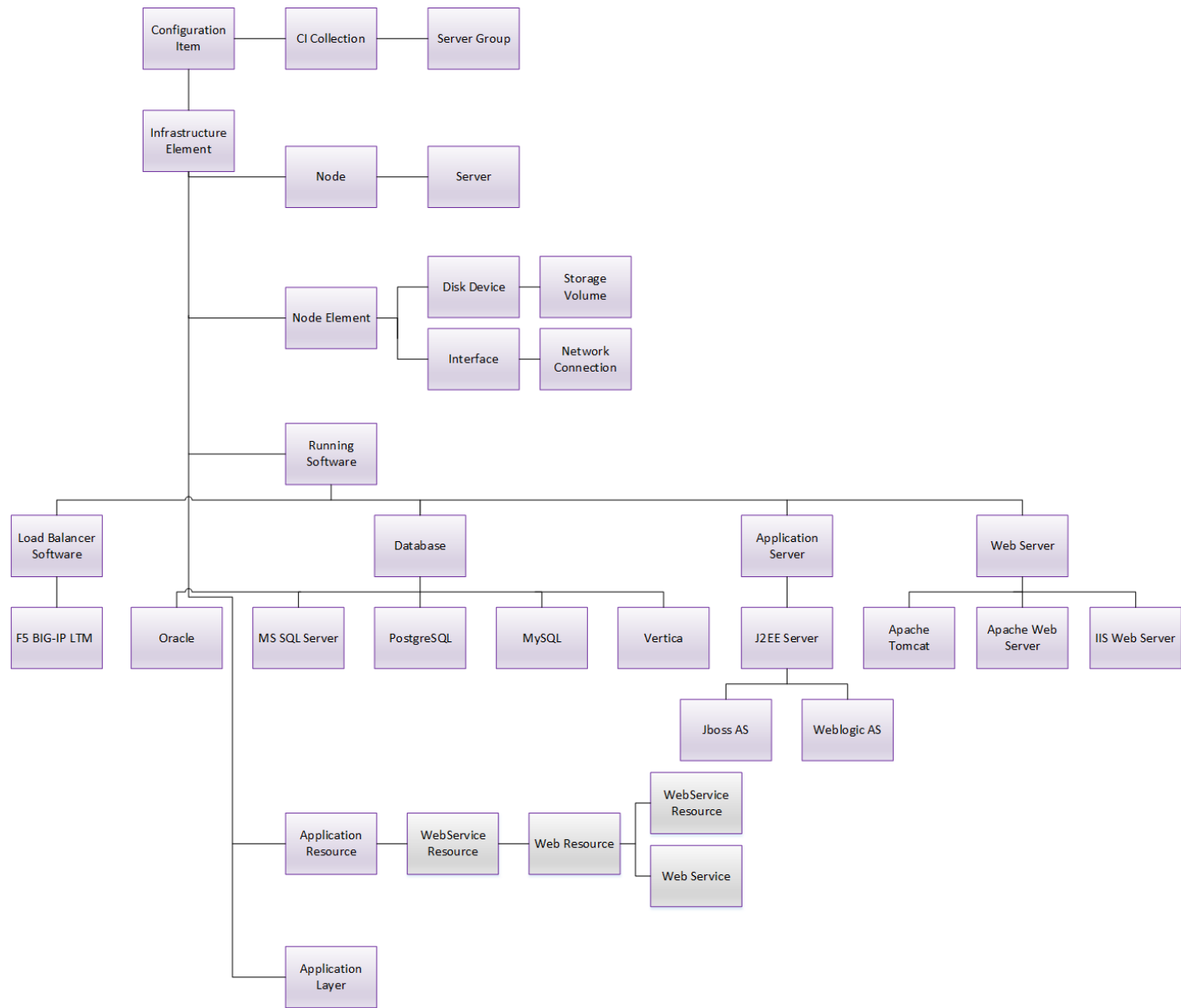
- **What will happen if the UCMDB server becomes unavailable during service provisioning? Will the service provisioning fail?**

No, the provisioning will not fail. The subscriptions will be active however, the topology in UCMDB will not be created.

Appendix

- [OOTB Components Hierarchy based on CI Types](#)
- [CI Types](#)
- [Reconciliation in UCMDB](#)

OOTB Components Hierarchy based on CI Types



CI Types

CI Types are Component Types specific for creating Sequenced Service Designs to integrate with UCMBD Provider. The CI Types are available in the palette CSA (CI Types based). The available CI Types in CSA are listed in the below table:

Component Type	Description	CI Type	Component Constraints	Resource Category Constraints
Configuration Item	Root Component for all components derived from CI Type or equivalent.	Configuration Item	None	None
Infrastructure Element	Contains infrastructure service components such as, individual servers or server groups that are required in a service design.	Infrastructure Element	None	None
Node	Represents a general purpose machine and is a base component type from which other component types representing devices such as, Virtual Machines will be derived.	Node	None	None
Server	Represents a general purpose machine which has an IP Address such as, Windows or Unix Servers.	Computer	Node Element Running Software	Application Compliance Compute Configuration Management IT Service Management Monitoring Service Usage Storage
CI Collection	Generic logical collection of components that are of type - Infrastructure Element or Running Software.	CI Collection	None	None

Component Type	Description	CI Type	Component Constraints	Resource Category Constraints
Infrastructure Service	It contains a collection of infrastructure service components, such as Server Groups, that are needed in a service design.	CI Collection	Server Server Group	Compute Configuration Management Infrastructure Monitoring Network Storage
Server Group	Represents a server group which contains one or more servers.	CI Collection	Server Running Software	Compute Configuration Management Infrastructure Monitoring Network Storage
Running Software	Represents the runtime aspects of a software that is running or is intended to be running on a server and is a base component type from which other component types representing various software will be derived.	Running Software	None	Application Monitoring
Database	Represents a database management system.	Database	None	None
Oracle	Represents an Oracle database server.	Oracle	None	None
MS SQL Server	Represents an MS SQL Server database server.	SQL Server	None	None
PostgreSQL	Represents a PostgreSQL database server.	PostgreSQL	None	None

Component Type	Description	CI Type	Component Constraints	Resource Category Constraints
Vertica	Represents a Verica Instance.	Vertica	None	None
MySQL	Represents a MySQL database server.	MySQL	None	None
Application Server	Represents a server that hosts an API to expose business logic and business processes for use by third-party applications.	Application Server	None	None
J2EE Server	Represents a J2EE application server.	J2EE Server	None	None
JBoss AS	Represents a JBoss application server.	JBoss AS	None	None
WebLogic AS	Represents a WebLogic application server.	WebLogic AS	None	None
Load Balancer Software	Represents a component type for all Load Balancer software.	Load Balancer Software	None	None
F5 BIG-IP LTM	Represents a F5 BIG-IP Local Traffic Manager software.	F5 BIG-IP LTM	None	None
Web Server	Represents a software that accepts http requests and delivers web pages as a response.	Web Server	None	None
Apache Web Server	It represents an instance of Apache Web Server.	Apache	None	None
Apache Tomcat	Represents an instance of Apache Tomcat web server.	Apache Tomcat	None	None
IIS Web Server	Represents an instance of IIS Web Server.	IIS Web Server	None	None

Component Type	Description	CI Type	Component Constraints	Resource Category Constraints
Application Resource	A base component type that is used to derive component types representing application resources.	Application Resource	None	None
WebService Resource	A base component type used to derive component types that represent web service resources.	WebService Resource	None	None
Web Service	A component type that represents a web service.	Web Service	None	None
Interface	A component type that describes a logical interface.	Interface	None	None

Reconciliation in UCMDB

IT has data barriers between organizations, sites and processes where management data remains segregated and could be limited in its value to IT. Most IT departments do not have a single authoritative source for key data that is reconciled against available data in the IT environment making it hard to:

- Predict effects of tactical and strategic decision.
- Aggregate IT data to information and aggregate IT information to business intelligence that drives IT decisions.

Reconciliation is the process of identifying and matching entities from different data repositories. The reconciliation engine is responsible for identifying and matching entities from different data collectors and storing them.

Many different data collectors can send CIs to UCMDB. In actuality, each different source might be providing information about the same CI. The reconciliation engine is responsible for identifying and matching entities from different data collectors and storing them without duplication.

Send documentation feedback

If you have comments about this document, you can send them to docs.feedback@microfocus.com.

Legal notices

Warranty

The only warranties for products and services of Micro Focus and its affiliates and licensors ("Micro Focus") are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Restricted rights legend

Contains Confidential Information. Except as specifically indicated otherwise, a valid license is required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright notice

© Copyright 2008 - 2018 Micro Focus or one of its affiliates.

Trademark notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

RED HAT READY™ Logo and RED HAT CERTIFIED PARTNER™ Logo are trademarks of Red Hat, Inc.

The OpenStack word mark and the Square O Design, together or apart, are trademarks or registered trademarks of OpenStack Foundation in the United States and other countries, and are used with the OpenStack Foundation's permission.

Support

Visit the Hewlett Packard Enterprise Software Support Online web site at <https://softwaresupport.softwaregrp.com>.

This website provides contact information and details about the products, services, and support that Micro Focus offers.

Micro Focus online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support website to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Access the Software Licenses and Downloads portal
- Download software patches
- Access product documentation
- Manage support contracts
- Look up Micro Focus support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require you to register as a Passport user and sign in. Many also require a support contract.

You can register for a Software Passport through a link on the Software Support Online site.

To find more information about access levels, go to <https://softwaresupport.softwaregrp.com/web/softwaresupport/access-levels>.

To check for recent updates or to verify that you are using the most recent edition of a document, contact your Client Director.