Codar

Software Version: 1.90

For Linux operating system

Use

Hewlett Packard Enterprise

Legal Notices

Warranty

The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise 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

Confidential computer software. Valid license from Hewlett Packard Enterprise 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

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

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

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

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.

This product includes an interface of the 'zlib' general purpose compression library, which is Copyright © 1995-2002 Jean-loup Gailly and Mark Adler.

Documentation Updates

To check for recent updates or to verify that you are using the most recent edition of a document, go to: https://softwaresupport.hpe.com/.

This site requires that you register for an HP Passport and to sign in. To register for an HP Passport ID, click **Register** on the HPE Software Support site or click **Create an Account** on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HPE sales representative for details.

Support

Visit the HPE Software Support site at: https://softwaresupport.hpe.com/.

Most of the support areas require that you register as an HP Passport user and to sign in. Many also require a support contract. To register for an HP Passport ID, click **Register** on the HPE Support site or click **Create an Account** on the HP Passport login page.

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

HPE Software Solutions Now accesses the HPSW Solution and Integration Portal website. This site enables you to explore HPE Product Solutions to meet your business needs, includes a full list of Integrations between HPE Products, as well as a listing of ITIL Processes. The URL for this website is https://softwaresupport.hpe.com/km/KM01702731.

Contents

Use

This topic gives information on usage of Codar.

- Create Designs

- Automate Release

Create Designs

Use the **Designs** area of the Codar Management Console to manage service designs, which are the recipes for automating the cloud. You can create the following types of designs:

- **Topology Design** - Topology designs specify components, relationships, groups, and properties. Use topology designs for Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) deployments that are enabled via Chef, Puppet, Server Automation and Operations Orchestration flow-based components. In contrast to sequenced designs, which more explicitly define the provisioning order and the sequence of actions that will run, topology designs are declarative in nature and do not include explicit actions or sequencing. The provisioning sequence is inferred by the relationships that exist between components in a topology design.

- **Sequenced Design** - Sequenced designs specify directed execution of service component lifecycles. Use sequenced designs for complex services and services that rely on run book automation, such as integrations with legacy data center systems. Create a sequenced design as a directed component hierarchy to define lifecycle execution. Sequenced designs use components to group multiple automation providers within a single entity. Sequenced designs permit explicit specification of lifecycle actions.

Refer to the following topics for more information:

- "Topology designs" on page 19

- "Sequenced Designs" on page 60

- "Resource Offerings Overview Tab" on page 132

- "View Resource Offerings" on page 143

- "Create a Resource Offering" on page 144

- "Import and Export a Resource Offering" on page 145

- "Select or Remove Providers with a Resource Offering" on page 149

- "Lifecycle Actions for Resource Offerings" on page 150

- "User Operations for Resource Offerings" on page 158

- "Custom Resource Offering Properties" on page 161

- "Resource Offering Categories" on page 169

- "View Component Palettes" on page 173

- "View Component Types" on page 180

Use
Create Designs

-

-

Create Designs

Use the **Designs** area of the Codar Management Console to manage service designs, which are the recipes for automating the cloud. You can create the following types of designs:

- **Topology Design** - Topology designs specify components, relationships, groups, and properties. Use topology designs for Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) deployments that are enabled via Chef, Puppet, Server Automation and Operations Orchestration flow-based components. In contrast to sequenced designs, which more explicitly define the provisioning order and the sequence of actions that will run, topology designs are declarative in nature and do not include explicit actions or sequencing. The provisioning sequence is inferred by the relationships that exist between components in a topology design.

- **Sequenced Design** - Sequenced designs specify directed execution of service component lifecycles. Use sequenced designs for complex services and services that rely on run book automation, such as integrations with legacy data center systems. Create a sequenced design as a directed component hierarchy to define lifecycle execution. Sequenced designs use components to group multiple automation providers within a single entity. Sequenced designs permit explicit specification of lifecycle actions.

Refer to the following topics for more information:

-

-

Topology designs

Topology designs are typically used for Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) deployments that are enabled via Chef, Puppet, Server Automation and Operations Orchestration flow-based components.

Topology designs are declarative in nature and do not include explicit actions or sequencing. The provisioning sequence is inferred by the relationships that exist between components in a topology design.

A topology design is composed of components. Each component can have a relationship with other components. For example, you might define a relationship between a server component and a networking interface component. Each component can also have properties, such as an IP address, password, or name. See "Components " on page 44.

When a topology design is associated with a release pipeline process, it is referred to as an application design. When it is associated to microservices, it is referred to as a service design.

Component types

A component represents one service design element that is required to deploy a service. A topology design can include concrete components and capabilities.

- A *concrete component* is defined in the design as a specific service, such as a vCenter server.

  A concrete component is deployed as defined and does not require composition.

- A *capability* is a component that can be fulfilled by any service that meets its criteria, such as a web server that could be fulfilled by either Apache HTTP Server or Apache Tomcat. Capabilities are identified in the designer by angled brackets, such as <<Server>>.

  A capability must be matched to an service design or a microservice to be fulfilled.

See "Topology composition" on page 26

Topology design versions

A version of a topology design is where you add components, define relationships, and enter property values for a design. Each version can be edited and configured as needed. Throughout the topology designs area, the version identifier is shown at the top of the window under the display name. See "Topology design versions" on page 33

Manage topology designs

Use the Topology Designs area to browse, view, and manage topology design.

Note: Only the admin user and Application Architect role have access to the Design sidebar menu item. Users with other roles can access designs in the Release Pipeline sidebar menu item if they have been granted access. See "Access control" on page 294.

Tasks

In the Topology Designs area you can perform the following tasks:

- **Manage design tags.** Click the **Manage Design Tags** icon in the lower left pane. See "Manage design tags" on page 29.

- **View designs.** In the left pane, select **All Designs** or select a user-defined tag, to display a list of the corresponding topology designs.

- **Create a new version.** Click **New** under the design name to create a new version. Each design may have multiple versions, and the most recently edited version is shown in bold. The most recent five versions of a design are shown beneath the design name, with links to individual versions. For designs with more than five versions, click **All** to see the complete list. See "Topology design versions" on page 33.

- **Create a new design.** See "Create, edit, and manage topology designs" on the next page.

- **Import a design.** See "Import or export a topology design" on page 30.

Create, edit, and manage topology designs

To create a topology design

1. Go to Topology Designs.
2. Click **Create** to create a new design.

To edit a topology design

1. Go to Topology Designs.
2. Select a design to view its details.
3. Click **Edit**.

Toplogy design properties

You can change the following properties for a new or existing design:

- **Palette** - For a new design only, select the palette for the design. See "Palettes" on page 229
- **Display Name** - Provide the display name for the design.
- **Description** - Provide the description for the design.
- **Initial Version** - Provide the version number you want to assign to the design.
- **Image** - Select an image that displays for the design. If you upload your own image, the supported file extensions include .jpg, .jpeg, .gif, and .png. The recommended image size is 256 by 256 pixels, and the image will be scaled to the appropriate size.
- **Tags** - Click **Select Tags** to choose from a list of tags that you can define to provide a structure for organizing and grouping the service designs. You can select more than one tag for a design. To remove a tag, hover the cursor over the tag you want to remove, and click the **Delete** icon.

Manage design as application design or service design

You can classify a selected topology design as either an application design or as a service design.

1. Navigate to **Designs** > **Topology design** > **Designer**.

2. Select a design to modify.

3. Click **Actions Menu** and select **Manage Release Automation**.

4. Select one of the following design classes:

- **Application design**, to manage the selected design as an application across lifecycle stages in the release automation.

- **Service design for design composition**, to make the selected design available as a service design or microservice design, when provisioning an application design for the selected lifecycle stages.

  If you select **Service design for design composition**, you can select the lifecycle stages in which the selected design must be available.

Note: You can select only one of the above design classes. If you select, **Application Design**, **Service design for design composition** will not be available for selection and vice versa.

Generate topology designs

Instead of manually creating a design from scratch by assembling components and creating OO flows, you can generate a topology design complete with all the components, OO flow, and package which is ready to deploy, using the Generate feature. This feature is only available for topology designs.

Roles

Only architect and admin users can generate designs.

Supported platforms

Codar supports generation of topology designs on machines with the following flavors of Unix and Microsoft SQL.

Unix: Both single-tier and two-tier designs supported with the following flavors:

- RHEL (7.x)

- Ubuntu (12.04 and 14.04)

Microsoft SQL: Only single-tier designs supported.

- Windows 2008 (64-bit machine)

- Windows server 2012 (64-bit machine)

For machines with Windows server 2012, the following configuration is necessary for auto generation:

1. Open **Run** prompt and enter 'gpedit.msc'.

2. Press **Enter**.

3. Navigate to **Computer Configuration>Windows Settings>Security Setting>Network List Manager Policies**.

4. Choose your network name in the right pane.

5. Go to **Network Location** tab and change the Location type from **Public** to **Private**.

To generate a topology design

1. From the Codar dashboard, go to **Designs** > **Topology** > **Designer**.

2. In the **All Designs** page, click the gear icon in the right pane and select **Generate Design**.

3. In the **Generate Design** dialog, specify the following details in the **General** tab:

Field	Description
Display Name	Name of the design you are generating.
Description	Description of the design you are generating.
Version	Design version. Example: 1.0
Options	Application tier type.

4. Click **Next** and specify the following details in the **Infrastructure** tab.

Field	Description
IP Address	IP address of the machine on which Codar is hosted.
Username	User name to access Codar.
Password	Codar password.
Private Key	Private key to access the machine (for Linux machines).

5. Click **Next** and specify the following details in the **Web Application** tab. Based on the application tier type selected, the following fields appear:

Application Tier Type	Field	Description
Single Tier J2EE Application on Linux	Application War Location Tomcat Binary Location	Location of the .war file. For local machines, specify the location of the machine on which Codar has been installed. Location of the Tomcat binary file. **Note:** If you leave this field blank, by default, Codar will install Tomcat via Linux packages. You can override it, if required.
Single Tier IIS Application on Windows	Application Location Application Name	Location of the application. Name of the application.
Two Tier J2EE Application on Linux	Application War Location Tomcat Binary Location MySQL Script Location	Same as above. Same as above. Location of MySQL script. By default, MySQL will be installed with following credentials:

Application Tier Type	Field	Description
		• User name: root • Password: 2the#Moon You must change the password later either manually or by running a script.

6. Click **Finish**.

Sample web application

You can download a sample web application from the following location:
https://tomcat.apache.org/tomcat-8.0-doc/appdev/sample/.

Save the downloaded .war file and specify this location in the **Application War Location** field.

Topology composition

Components are the building blocks for your designs. Some components are created when you install Codar, and you can add new components in the Components sidebar menu item in the Designs area.

A component represents one service design element that is required to deploy a service. A topology design can include concrete components and capabilities.

- A *concrete component* is defined in the design as a specific service, such as a vCenter server.

  A concrete component is deployed as defined and does not require composition.

- A *capability* is a component that can be fulfilled by any service that meets its criteria, such as a web server that could be fulfilled by either Apache HTTP Server or Apache Tomcat. Capabilities are identified in the designer by angled brackets, such as <<Server>>.

  A capability must be matched to an service design or a microservice to be fulfilled.

See "Components " on page 44.

You can use these components to create the following types of topology designs:

- A *concrete design* is any design that contains no capabilities. It may include components that support a capability, but it cannot directly contain capability components.

- A *partial design* is any design that contains capabilities. Such a design cannot be provisioned on its own and requires the selection of a compatible service design.

- A *service design* is a design that can fulfill all capability requirements of a partial design.

- A *microservice* is a design that can fulfill one part of a capability.

- The combination of a partial design with a service design creates a *composite design* that is itself a concrete design. A composite design is normally hidden from view in the list of topology designs, but it can be made visible by saving the design from a test run. See "Test a topology design" on page 40.

When only concrete components are included in a design, those specific services are provisioned as specified. When a design contains capabilities, the design requires a separate design that satisfies each capability.

The separate design can be a service design with concrete components or a set of microservices that together fulfill the capability requirements. If a microservice satisfies more than one capability, the topology instance can be shared between two or more requirements, or each requirement can use a new instance of the microservice.

For example, this design contains a My application component that has relationships to <<Application Server>>, <<Database Server>>, and <<Web Server>> (each of which is a capability):
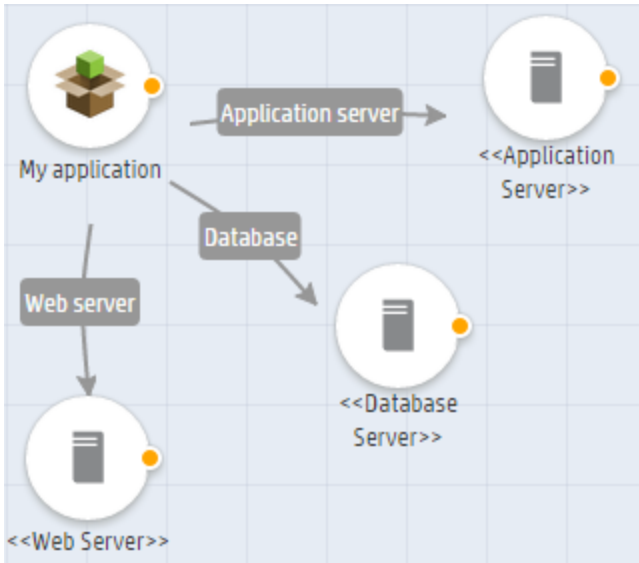
A suitable service design might contain PHP, MySQL, and Apache components. A microservice would contain only one of those components, such as a MySQL server.

Designs with a capability cannot be published but can be provisioned as a test run. See "Test a topology design" on page 40.

When you deploy a package from a partial design, you select an service design or microservices that meet the requirements of the design.

Requirements (required characteristics)

You can specify requirements for capability components. A requirement allows you to refine which concrete designs should be selected when a partial design is provisioned. For example, Apache HTTP Server may be a required characteristic that is specified on a Web Server capability, so only designs that include Apache Web Server as a supported characteristic of the component will be selected when a design is provisioned. See "Component characteristics" on page 58.

Testing a partial design

During the test run of a partial design, you will be prompted to select from the list of candidate service designs, each of which meets the following criteria:

- Is a concrete design.

- Contains concrete components that support each of the capabilities used in the partial design, each of which supports all of the required characteristics specified on the capability in the partial design.

- Has property values configured for all required properties in the design.

- Is not a composite design that was created by the combination of a partial design with another concrete design, unless that composite design was subsequently saved as its own design (see the description of **Composite design** below for more information).

- Has no validation errors.

For example, consider a partial design that includes a concrete component My App with relationships to Web Server, Application Server, and Database Server components.

Assume the Web Server, Application Server, and Database Server components have required characteristics of Apache 2.4, PHP 8, and MySQL 5.5, respectively.

The partial design could be provisioned with a that contains Apache Web Server, PHP, and MySQL components, respectively supporting the Web Server, Application Server, and Database Server capabilities, which support the characteristics Apache 2.4, PHP 8, and MySQL 5.5, with each component configured to be provisioned on a server.

See "Test a topology design" on page 40.

Manage design tags

Tags are user-defined, color-coded labels and images used to organize and group topology designs. A tag can be assigned to multiple designs.

To manage design tags

1. Go to Topology Designs.

2. Click the **Manage Design Tags** icon in the lower left pane.

3. Click the **Create Tag** icon to add a new tag.

4. Click the **Edit Tag** icon to change the selected tag.

5. Click the **Delete Tag** icon to remove a tag.

6. Click **Done** when you are finished.

Import or export a topology design

The export process creates either a zip file with all the topology files, design, and dependencies as either a zip or JSON file. This file can then be imported to Codar.

The import process imports archives of designs and their supported artifacts. Designs with the same internal name are considered to be functionally equivalent and are not imported. When you import a version of an existing design, the imported design is saved as a version of the existing design.

During the update process, identical artifacts that exist on the target system are updated (overwritten) with the changes from the archive. Artifacts are created if they do not exist on the target system.

This process imports all artifacts present in an archive, whether or not they exist on the target system. During this operation, if there is an artifact with the same internal name in the system, the name, the display name, and the description of the artifact are modified internally; the display name and the description are appended with "Superseded on" and the date. The internal name, display name, and description of the artifact being imported remain intact.

Exporting a design creates a zip or JSON file for the design you are exporting, which contains associated artifacts, icons for customizing the artifacts, and the Manifest document, which contains meta-information about the archive files.

When you import a design, any tags you have selected in the designs area do not affect the imported design. The imported design will contain the tags that were included when the design was exported, and new tags will be created on the system, as necessary, to match what was exported.

If you import a design that already exists, access control for the existing design is retained. If you try to import a design and you do not have access to the design, then the import will not be permitted.

> **Note**: Importing and exporting JSON files is recommended for infrastructure as a service (IaaS) applications. For other uses, we recommend that you export designs as a zip file.

Prerequisites

Before you import a design archive, complete the following prerequisites:

- You can only import JSON files using the API.

- If you import a JSON archive and the design has a dependency on any custom component types, then these custom component types must be imported before the design can be imported.

- When you export a zip file, the export can only reference files or content contained within the file itself, or that are already contained in the csa.war file. By default, all images in the archive files

must end in one of the following suffix values. For information on adding additional suffix values, see the *Codar Configuration Guide*.

jpg, jpeg, jpe, jfif, svg, tif, tiff, ras, cmx, ico, pnm, pbm, pgm, ppm, rgb, xbm, xpm, xwd, png, gif, bmp, cod, ief

To import a design archive

1. In the lower, right pane of the Topology Designs area, click **Import**.

2. Select or specify the zip file that contains the design you want to import. Archive file names for designs begin with SERVICE_DESIGN_.

3. Select an option:

   Import - imports new designs; does not update existing service designs. Note that you cannot import a design with the same internal name as an existing design.

   Update - imports new designs (and associated resource offerings) and updates (overwrites) existing designs. Check Preserve Originals to create backup copies of the original items, appending "Superseded on" and the date to the artifact display names and descriptions.

4. Click **Preview** to see a report of prospective results for the import process, including information about the artifacts and their status.

5. Click **Import**.

6. Click **View Detailed Report** to see a summary and details of the import process, including information about the artifacts and their status.

7. Click **Close**.

To export a design archive

1. Open the version of a design that you want to export.

2. In the Overview tab, click **Export**.

3. Save the exported design, if required by your browser.

When you export a design archive, the design and all referenced resource offerings are packaged in an archive file with the following name:

SERVICE_DESIGN_<service_design_display_name>_<service_design_id>.zip

To export a design JSON

1. Open the version of a design that you want to export.

2. In the Overview tab, click **Export**.

3. Select JSON and click **Export**

4. Save the exported design JSON, if required by your browser.

When you export a design JSON, the design and all its components values are added in the JSON file with the following name:

<service_design_display_name>.json

Topology design versions

A version of a topology design is where you add components, define relationships, and enter property values for a design. Each version can be edited and configured as needed. Throughout the topology designs area, the version identifier is shown at the top of the window under the display name.

For general information about topology designs, see "Topology designs" on page 19.

Tasks

- **View versions of a design.** The versions list is sorted alphanumerically. You can click a version identifier to open that version of the design. See "Manage a topology design version" on the next page.

- **Create a new version.** While viewing the list of designs, click **New** under the design name. While viewing a design, open the Versions tab and click **New Version**. While viewing a version of a design, click **New Version** to create a new version based on the current design. Each design may have multiple versions.

  You may select a palette for the version. See "Palettes" on page 229.

  You can enter any version number or string and an optional description.

Manage a topology design version

The Overview tab for a version of a design provides a summary view of the topology design details.

Tasks

In the **Overview** tab, you can perform the following tasks, although some tasks may not be available depending on your role:

- **Edit this version.** See "Manage topology designs" on page 20.

- **Publish this version of the design.** Published designs are indicated in the Overview tab by a Published icon in the upper right corner. See " Publish a design version" on page 79.
  - You can publish a design only if it has no validation errors.

  - A published design cannot be modified, but you can use **Save As** or **Create New Version** to create editable designs based on the published design.

  - Partial designs contain capabilities and require composition with a compatible concrete design before they can be provisioned. Partial designs cannot be directly published. These designs are indicated in the Overview tab by a Requires Composition icon in the upper right of the window.

  - Published designs cannot be unpublished.

- **Save this version with a new name.** Use **Save As** to create copy of the current version of the design.

- **Create a new version.** This will create a new version based on the current version.

- **Export a version.** See "Import or export a topology design" on page 30.

- **Delete this version.** Deletes the selected version of the design.

- **Manage packages.** This button takes you to the design in the Release Pipeline area. See "Release pipeline" on page 248.

Best practices

- Perform a **Test Run** before creating a package.

- You must use **Save As** or **New Version** to modify a design that has a package.

- Deleting a design cannot be undone.

Topology designer

A version of a design has a Designer tab, where you create a layout of components and their relationships.

A single design normally contains components of multiple provider types (for example, VMware vCenter and HPE SiteScope components); however, if you include Helion OpenStack® in a design, then you can only use HPEHelion OpenStack® components.

Tasks

Designs can be created from the set of topology components that are defined in the Components area of the Cloud Service Management Console. Many components are included with Codar. If necessary, import any additional components that you need prior to creating a design. See "Import components" on page 46.

> **Note**: Designs are saved automatically every five seconds after a change or immediately after a significant and meaningful change is made to the design.

- **Add components to your design.** From the set of components (based on the palette you selected for the design) in the left pane of the Designer, drag-and-drop or double-click the component you want to add. The components you select can be concrete components or capabilities. Capabilites are identified by angle brackets, such as <<Server>>.

- **Add a relationship between components.** Click the orange dot on the right side of a component and drag to establish a relationship with another component. Components that are not appropriate will be grayed out.

  You can also click a component select the **Add** icon to see a list of compatible component relationships, then select a component to add to the design.

  You can create relationships only between components that have been configured to allow such relationships. In addition, the direction of the relationship (outgoing or incoming), and the number of allowable relationships are configured per component. See"Component relationships" on page 51.

  After the connection is made, the arrow automatically points in the proper direction, as defined by the component relationship configuration. Note that capabilities do not support outgoing relationships when included in a design; capabilities support only incoming relationships.

- **Remove a relationship between components.** Click a relationship arrow, then click the delete icon to delete the selected relationship.

- **Edit a component.** Select a component and then click the **Edit** icon. You can change its name and the group it belongs to.

- **Delete a component.** Select a component and click the **Delete** icon.

- **Set property values for components and relationships.** Select a component and its properties will be displayed in the panel on the right.

  Properties on a component can be set to modifiable during package deploy or redeploy. All properties can be modified during a test run.

  Properties on a component or relationship can set to modifiable during service modification to indicate that the property value can be modified during service modification. This check box is disabled if Modifiable during service creation is not selected for the property in the Properties tab of the Components area.

  Click **Save** to save your changes.

- **Select requirements for a capability.** Select a capability and then click Requirements in the panel on the right.

  Click **Select** to add or remove characteristics. If you filter by applicable characteristics, the list includes all characteristics supported by all concrete types that support the selected capability. For example, if there are concrete components Apache 2.2 Web Server and Apache 2.4 Web Server that support the Web Server capability, all characteristics supported by those concrete types will be displayed.

- **Create, edit, or delete a group.** See "Topology design groups" on page 38.

- **Assign components to a group.** See "Topology design groups" on page 38.

- **View validation errors.** Validation is a continuing process during design creation, configuration, and editing. Designs with validation errors can be saved but not published. Hover over an error icon to see more information about the validation error. Common sources of validation errors include required properties on components that are undefined, undefined groups, and required relationships that are not configured.

Best practices

- Partial designs contain capabilities and require composition with a compatible concrete design before they can be published and deployed.

- Use the icons at the bottom center of the Designer tab to use automatic layout or to zoom in and out.

- Click the arrow to open and close the component drawer if you need more space in the Designer window.

- Some features are not available for published designs; published designs are displayed in read-only mode.

Topology design groups

Groups are used to create scalable stacks of components in the Designer tab. The scalable stacks are made up of related components that can be scaled out together. For example, you might create a web server group that contains the following three components in its scalable stack:

- A web application

- An Apache web server

- The server component on which the other two components are hosted

The groups created for a topology design are unique to that design; they are not shared across designs. You can create a relationship between two components inside a group or between a component outside a group and a component inside a group, but you can't create a relationship between components that belong to different groups.

When you deploy a package from a partial design with a scalable group, you select a service design that has a group which meets the requirements of the design.

Manage groups

In the Designer tab, click **Manage Groups** in the lower left, which allows you to perform the following tasks for groups:

- **Create a group.** Click the create icon, and provide the following information for the group:

  Display Name - Type a unique display name for the group.

  Image - Select an image that will be displayed for the group.

  Color - Select a color that appears as the background for the group when it displays in the Designer tab.

- **Edit a group.** Click the edit icon and edit the display name, image, or color for a group.

- **Delete a group.** Select the group you want to delete and click the delete icon.

- **Move a group.** Click inside a group and drag the group to a new location.

- **View unused groups.** Information for unused groups is displayed in the lower left corner of the designer when a topology design contains an unused group. Unused groups are groups that have been defined but that do not contain components. Click **Show** to see the list of unused groups and click **Hide** to close the list.

Add components to a group

1. In the Designer tab, click a component in the design to access the menu that slides out to the right of the component. Click the edit icon, and select a group to which the selected component will be added.

2. Create relationships between components in the group. A group is displayed in the designer as a colored rectangle, which contains all the components in the group.

   You can create relationships between components that are part of the same group but not between components in different groups. See "Topology designer" on page 35.

3. Configure properties for the group. Click in the group's colored background or title bar, and do the following:

   a. In the Properties tab, in the right pane of the designer, set **Instance Count** to indicate the number of instances of the group that will be provisioned for the topology design.

   b. Set properties to **Modifiable only during package deploy** to indicate that the property value can be modified when a package is deployed or redeployed. All properties can be modified during a test run.

   c. Set properties to **Modifiable during package redeploy** to indicate that the property value can be modified during package redeploy.

   d. Click **Save**.

Test a topology design

Use the **Test** tab to test and manage topology design provisioning. You can test the following types of topology designs:

- **Partial design** - During the test run of a partial design, you will be prompted to select from the list of designs or microservices.

- **Concrete design**

See "Topology composition" on page 26 for information about these types of designs.

Tasks

You can view the following information in the **Test** tab:

- The name of the test run.

- Date and time of the test run.

- Current status of the test run.

- The user who initiated the test run.

You can perform the following tasks in the **Test** tab:

- **Cancel** - Cancel the realized design so that the instance is terminated. You must cancel before you can delete.

- **Delete** - Deletes a canceled or failed deployment instance. You must cancel before you can delete.

- **Test Run** - See Test Run Wizard below.

- **Refresh** - Refreshes the data in this tab.

You can link to detailed results of test provisioning by clicking the name of the test run. You can see the following information and perform the following tasks:

- **Overview** tab shows details of the test run.
  - **Cancel** the realized design so that the instance is terminated. You must cancel before you can delete.

  - **Delete** a canceled or failed deployment instance.

- **Events** tab shows:
  - ○ **Event Time** - indicates the time the event occurred.

  - ○ **Lifecycle State** - indicates the state of the event execution, for example, Deploying or Undeploying.

  - ○ **Action** - events, such as deployment, server restart, etc., that are executed during the test run.

  - ○ **Source** - the design component that is the source of the event.

  - ○ **Status** - current state of the test run. If you click on an event you see details about the event in the Overview tab and property values used in the test run in the **Properties** tab.

- **Topology** tab shows:
  - ○ The topology diagram, properties, status of the test, and an instance summary.

  - ○ Select a server component in this tab to see properties in the right panel and available actions below. You can invoke an action to verify that it is working.

- **Providers** tab shows which provider instances have been used for deployment.

Test run wizard

Use **Test Run** on a topology design's Test tab.

The **Test Run** wizard allows you to specify the following:

- **General** - Set properties that are common to all designs.
  - ○ Display Name - Type a display name or use the system proposed name of `Test run of <design name>`.

  - ○ Environment - You can select an environment to restrict provider selection to only those providers located in the selected environment. Optional.

- **Service Designs** - Select a design from the list that, together with the partial design, will form a composite design that can be provisioned. If you select **Custom Selection**, you can select a microservice for each requirement in the partial design. Only service designs or microservices that fulfill the requirements are available.

- **Group Properties** - Properties for groups in the design.

- **Component Properties** - Properties for components in the design.

- **Relationship Properties** - Properties for partial designs. The list includes concrete designs that support all the capabilities and required characteristics in the partial design.

Best practices

- If you cancel a deployment, the test run is still listed in the tab. If you no longer need the entry, you can delete it.

- Select a new name for every test run so you can distinguish between them.

- Test your designs using various deployment scenarios and property configurations to ensure design stability before you create a package.

Environment groups

You can assign different environment groups based on the lifecycle stage for a design. When the package is deployed, an environment is chosen from the group.

For example, you may want to restrict deployments in the Testing stage to certain environments. You would create an environment group that includes only the environments that are appropriate for this lifecycle stage. You would exclude environments where you don't want a test deployment created.

By default, a package can be deployed to all environments unless the design contains an environment association.

To change the environment groups for a lifecycle stage

1. Go to the Environments tab for a design.

2. Click **Manage Environments** to the right of the lifecycle stage.

3. Add or remove resource environments.

4. Click **Save**.

Components

Each component binds to a single provider for fulfillment automation. The component lifecycle provisioning is delegated to the providers.

Use the **Components** area in topology designs to import, create, and configure components. Codar provides a number of out-of-the-box components you can use for creating topology designs.

Concrete components

Concrete components are the most common components that you will interact with.

- Concrete components can contain properties, relationships, and operations.
- Concrete components can express support for capabilities, such as server, application server, database server, and web server, and can also support characteristics.
- Concrete components can be included in a design.

Abstract components

Abstract components provide a base type for other components and can be the target of relationships configured on a component.

- Abstract components can contain properties and relationships.
- Abstract components cannot express support for capabilities or characteristics.
- Abstract components cannot be included in a design.

Capability components

A capability component is used in a partial design and only contains properties or relationships. For example, a capability component might contain the properties required for an application server. It is associated with a concrete component that supports the capability. When a concrete component supports a capability, the concrete component inherits the capability's relationships and must provide property mappings from the concrete component properties to the capability properties.

- Capabilities can be the target of relationships configured on a component.

- Capabilities can be included in a design, but for such a design to be successfully provisioned, another design must exist that contains a concrete component supporting the capability.

Tasks

You can perform the following tasks in this area:

- **Search for components.** Enter your search string in the **Search** field.

- **Select how to view components.** Select to view: **By Palette**, **By Tag**, or **By Provider Type**. Click an item in the left pane to see its components displayed in the right pane.

- **Manage tags.** When you are in the **By Tag** view, click the **Manage Component Tags** icon in the lower left pane. Tags are labels that provide a structure for organizing and grouping related items. A component can be assigned to multiple tags.

- **Manage palettes.** When you are in the **By Palette** view, click the Manage Palette icon in the lower left pane. See "Palettes" on page 229.

- **Navigate to component details.** Select a component in the list to see more information about it.

- **Import components.** See "Import components" on the next page.

- **Create a topology component.** Click **Create** and provide the following information:

  Display Name - Type a unique display name for the component.

  Description - Type a description of the component.

  Version - Type a version number for the component.

  Functional Type - Select a functional type for the component. The functional type can be Abstract, Concrete, or Capability. See the descriptions in the Concepts section above for more information about the functional types.

  Provider Type - Select a provider type for the component.

  Image - Select an image that will display for the component.

  Tags - Select one or more tags that will include the component. Tags are user-defined, color-coded labels and images used to provide a structure for organizing and grouping topology components.

Best practices

If you have a large number of components, use tags to organize them into meaningful groups.

Import components

Importing provides you with the ability to use content from sources external to Codar. Importing is the only way to add additional components, as you cannot create components manually.

Importing components is done through an import wizard, with options based on the selected **Import Source** (provider type). The provider types are:

- Puppet
- Chef
- Operations Orchestration
- Server Automation

Importing components

From the Topology Components area, click **Import**, and provide the following information:

Item	Description
General	• **Import Source** - Select the provider type to use as the import source for the component. • **Provider Instance** - Select the provider instance that contains the component you want to import. Provider instances must first be configured in the Providers area to enable import for Chef, Server Automation, and Puppet provider types. For Operations Orchestration, content can be imported from the instance that is configured to integrate with Codar. • **Image** - Select an Image that will display for the component. • **Tags** - Select one or more tags that will include the component. Tags are user-defined, color-coded labels and images used to provide a structure for organizing and grouping topology components.
Select Content	Depending on your selections in the General tab, select the content for the component: • **Chef** - Select one or more Chef cookbooks to be imported as new Codar topology components. Each selected cookbook will create a separate component. • **Operations Orchestration** - Browse the Operations Orchestration library and select one or more standard component elements that will be imported as new topology components. Each selected item will create a separate component. • **Server Automation** - Select one or more Server Automation policy to be imported as new topology components. Each selected policy will create a separate component. • **Puppet** - Select one or more Puppet classes to be imported as new topology

	components. Each selected class will create a separate component.
Summary	Review your import selections, and click **Import**. The imported components will appear in the Topology Components list.

Component overview

You can view a summary of information about the selected component in the **Overview** tab.

Tasks

You can perform the following tasks in this area:

- **View a component.** See the following component information:
  - **Display Name** - The name displayed for the component.

  - **Description** - The description configured for the component.

  - **Version** - The version configured for the component.

  - **Provider Type** - The provider type configured for the component.

  - **Functional Type** - The functional type can be abstract, concrete, or capability. See "Components " on page 44.

  - **Image** - The image configured for the component.

  - **Tags** - The tags configured for the component.

- **Edit a component.** The following restrictions apply to the editing of components:
  - For some out-of-the-box components, you can edit the **Display Name**, **Description**, **Image**, and **Tags**.

- **Save a component with a new name.** Save the component with a new **Display Name** and **Description**. **Save As** cannot be used for abstract components or capabilities; it can be used only for concrete components.

- **Delete a component.** You cannot delete a component that is being used in a topology design. Some out-of-the-box components cannot be deleted.

Component properties

Properties provide a base set of attributes that can be used and edited when creating components in a service design. They represent configuration settings to be applied to the component during service design deployment. The value defined for a component property is the default value exposed in the service design.

Tasks

View the following information about component properties:

- **Display Name** - The display name of the property.

- **Value** - The default value for the property.

- **Visible** - This label displays if the property is visible to the designer. If this label does not display, the property is not visible to the designer.

- **Supports Modify Lifecycle** - This label displays if the property can be modified during the modify lifecycle transition phase.

Create a component property:

- You cannot create properties of a component that is an abstract component or a capability.

- Newly created properties do not affect existing service designs. You must manually update any design that uses the component by recreating the component in the design.

- For components that are being used in a topology design, you can only add additional properties, and the new properties cannot be **Required**. You cannot modify existing properties, except to change the default values for **Display Name**, **Description**, and **Default Value**.

Click **Create** and enter the following information for a property:

Item	Description
Type	**Boolean** - True or False. **Integer** - A positive or negative whole number or zero. **List** - A list of string values. **String** - A sequence of characters.
Name	A unique name for the property. This value cannot be changed after the component is created.

Item	Description
Display Name	The name that displays for the property.
Description	The description that displays for the property.
Resource Type and Unit for a Measurable Property	Available for **Integer** properties only. Select a resource type and unit (other than **None**) to make this property measurable and to enable automatic accounting of resource usage in a provider's resource pool.
Default Value	The default value exposed in the service design.
Confidential Data	Select this box to mask the values so that they cannot be read in the user interface; no encryption of the value is performed.
Consumer Visible	Use the check box to indicate if the property will be visible in the Marketplace Portal.
Designer Visible	Use the check box to indicate if the property is visible in the topology design.
Required	Use the check box to indicate if the property is required in the topology design. Required can only be selected for **Designer Visible** Properties

Edit a component property:

You cannot create properties of a component that is an abstract component or a capability.

Click the gear icon and select **Edit**.

Delete a component property:

You cannot delete properties that are being used in a design.

Click the gear icon and select **Delete**

Component relationships

Relationships in topology designs define dependencies between components and also impact how a design is provisioned. For example, imported Chef components require a server in order to be provisioned. Therefore, all imported Chef components are created with an outgoing relationship to the server capability, ensuring that a server is provisioned before the Chef component.

When importing new components, you may need to define new relationships. Some relationships are created for you automatically at import time, such as the Chef component dependency on server discussed above. If there are additional dependencies, define them manually in this tab.

Tasks

View the following information about the relationships:

- **Direction** - Outgoing or Incoming.

- **Target or Source Component** - Click the component name of the target or source component to open the **Overview** tab for that component.

- **Display Name** - The display name of the relationship.

Use the corresponding icon or button to:

- **Create or edit a relationship.**
  - You cannot create or edit incoming relationships.

  - You cannot edit a component that is an abstract component or a capability.

  - After a component is used in a design, you cannot add a new *required* relationship or make an existing relationship required. You can only change its display name and description.

    Provide the following information for a relationship:

Item	Description
Target Component	Select the target component for the relationship from the **Select Component** dialog, which provides a list of available components.
Display Name	The name for the relationship.
Description	The description for the relationship.
Maximum Outgoing Relationships From	The number of instances of the target component that a single instance of the source component can be connected to.

Item	Description
Source	
Maximum Incoming Relationships To Target	The number of source components that can be connected to a single instance of the target component.
Required Relationship	Use the check box to indicate if the relationship is required in topology designs.

- **Delete a relationship.** Click the delete icon. You cannot delete incoming relationships.

Component operations

View the following information about the operations:

- **Display Name** - The display name of the operation.

- **Parameters** - The parameters associated with the operation.

- **Visible** - This label displays if the operation is visible to the user. If this label does not display, the operation is not visible to the user.

Use the corresponding icon or button to:

- **Import** an operation. Click **Import**. You can import an operation, assign a lifecycle action (for example, Deploy or Undeploy), and edit operation parameter mappings. For example, you can import a new Operations Orchestration operation into an existing Chef component.

  To import, select one or more Operations Orchestration flows to import as new operations. Each item you select will create a separate operation on the topology component

- **Edit** an operation. Click the gear icon for the operation and select **Edit**.

You cannot edit a component that is an abstract component or a capability.

Complete all steps in the Edit Operation wizard.

- **Delete** the operation. Click the gear icon for the operation and select **Delete**.

Provide the following information for an operation:

Item	Description
General	**ID** - A unique identifier for the operation. You cannot edit this field. **Display Name** - The name that displays for the operation. **Description** - The description that displays for the operation. **Visible** - Use the check box to indicate if the property is visible. **Lifecycle Action** - Select a lifecycle action that will be used during provisioning or de-provisioning: **Deploy** – Actions specified in this lifecycle phase realize a component during the provisioning of a design that is using the component. **Deploy Failure Handler** – Actions specified in this lifecycle phase handle failures that occur during the execution of the **Deploy** lifecycle phase action on a component as part of provisioning a design. **Undeploy** – Actions specified in this lifecycle phase de-provision a component during the de-provisioning of a service instance that is using the component.

Item	Description
	○ **Undeploy Failure Handler** – Actions specified in this lifecycle phase handle failures that occur during the execution of the **Undeploy** lifecycle phase action on a component as part of de-provisioning a design.
	○ **Modify** – Actions specified in this lifecycle phase are executed on a component during the modification of a realized component in a service instance, when the service instance is modified by the **Modify Subscription** operation.
	○ **Modify Failure Handler** – Actions specified in this lifecycle phase are executed for the cleanup of failed **Modify** actions on a component during the modification of a service instance.
	○ **Unmodify** – Actions specified in this lifecycle phase revert the effect of modification on a successfully modified component. This action occurs during the rollback of an unsuccessful overall modification transition of the service instance due to failure occurring on another component that is also subject to modification. The result is that the service instance reverts to the state it was in prior to the modification operation.
Parameters	• **Input Parameter Mappings** - The list of input parameters and their configured parameter mappings for this operation. Click the appropriate icon to edit or delete an input parameter. See below for more information about editing.
	• **Output Parameter Mappings** - The list of output parameters and their configured parameter mappings for this operation. Click the appropriate icon to edit or delete an output parameter. See below for more information about editing.

Create or Edit Parameter Mapping	**Create or Edit Parameter Mapping**
	
	To create a parameter mapping, click the **Add Parameter** button just below the **Input Parameter Mappings** or **Output Parameter Mappings** list. Typically, you will add a parameter if the operation is tied to an Operations Orchestration flow and you have modified the flow to have new inputs or outputs.
	
	To edit an existing parameter mapping, click the **Edit Parameter Mapping** icon.
	
	• **Name** - A unique name for the parameter.
	
	• **Display Name** - The name that displays for the parameter.
	
	• **Description** - The description for the parameter.
	
	• **Required** - Use the check box to indicate if the parameter is required.
	
	• **Mapping Type** - Select the mapping type for the parameter mapping.
	  ○ **Not Mapped** - Select to provide no parameter mapping.
	
	  ○ **User Value** - Select to prompt the user for a value during design execution. This mapping is valid for public actions only. When a public action is executed, the user is asked to provide values for all parameters mapped to the **User Value**.
	
	  ○ **Component Property** - Select to map the parameter to or from a property on this component. Select the component property in the **Value** list.

Item	Description
	○ **Constant Value** - Select to provide a constant value for the parameter. Enter the constant value in the **Value** field. This is not available for output parameters. ○ **Multiple Properties** - Map to multiple properties. Click **Add Parameter** to create additional parameters. ○ **Provider Property** - Select to map the parameter from a property on the resource provider that is used to provision the component. Enter the name of the resource provider property in the **Value** input field. ○ **Relationship Target Property** - Select to map the parameter from a property on a different component that this component has a relationship to. Select the proper relationship in the **Relationship** list, and then select the component property in the **Value** list.

Component capability

See "Components " on page 44 for a definition of capability components.

Concrete components can claim support for a capability component. A supported capability includes a reference to the capability component, as well as property mappings between the capability component and the concrete component. A single capability, such as Web Server, can be supported by multiple concrete components; however, a concrete component can only support a single capability.

A capability component can be included in a design to indicate that a concrete component supporting that capability is required in order to successfully provision the design. Designs that include capabilities are referred to as partial designs and require composition with another design in order to be provisioned. For additional information on partial designs, see "Test a topology design" on page 40.

Tasks

View the following information about the capability:

- **Supported Capability** - The display name of the supported capability, if any. Supported capabilities are provided out-of-the-box for some concrete components.

- **Property mappings** - A list of each **Capability Property** and the **Component Property** from which it gets its value.

> **Note:** To use **Provider Type** mapping property in the **Operations** tab, for Parameter Mapping, you must associate the component with a provider type.

Perform the following tasks:

- **Refresh the data in this tab.**

- **Add or Edit a supported capability.** You cannot add or edit a capability if the component is being used by a topology design. Click **Add Supported Capability** or select a supported capability and click **Edit**. Provide or modify the following information:

Item	Description
Selected Capability	Choose a capability to add to this concrete component. You cannot edit this field after a supported capability has been added.
Property Mappings	For each **Capability Property** in the list, do one of the following: - Select a **Component Property** from which the **Capability Property** gets its value. - Leave the **Capability Property** as **Not Set**.

Item	Description
	Note that any capability property that has the same 'property Name' as a concrete component property (using a case-sensitive string match) is automatically mapped.

- **Remove the selected capability.** You cannot remove a capability from a component if the component is being used by a topology design.

Component characteristics

The **Characteristics** tab is visible for concrete components.

Characteristics describe a component and must be used in combination with capabilities. For example, an Apache 2.4 Web Server concrete component may define support for the following characteristics: Apache Web Server, Apache Web Server 2.x, or Apache Web Server 2.4.

If this component supports the Web Server capability, a service designer can include a Web Server capability in a design and specify that the design requires the Apache Web Server 2.4 characteristic. This allows the Apache 2.4 Web Server component (and any other Web Server components supporting this characteristic) to be chosen as part of provisioning the component.

A component should be associated with a capability when supported characteristics are configured. For more information, see "Component capability" on page 56.

Tasks

View the following information about the characteristics:

- **Supported Characteristics** - See a list of the characteristics supported by the component.

Perform the following tasks:

- **Select characteristics** - Click **Select**. Or, if no characteristics are configured for the component, click **Select characteristics for this component**. See the following table for selection information:

Item	Description
Available characteristics	Select one or more characteristics in the list, and click **Add**.
Selected characteristics	View the list of selected characteristics, or select one or more and click **Remove**.
Manage characteristics	**Create a characteristic** - If this is your first characteristic, click **Create your first characteristic now**. - Provide a **Display Name** for the characteristic. - Provide a **Description** for the characteristic. When characteristics already exist, you can: - Click the create icon to create a new characteristic. - Click the edit icon to edit a selected characteristic.

Item	Description
	• Click the delete icon to delete a selected characteristic. You cannot delete a characteristic that is a required characteristic in a partial design.

Sequenced Designs

A sequenced service design is a blueprint for an orderable service. Each design includes service components created from component types or component templates that define what is automatically provisioned. You can use service designs to create a complex set of automated elements that support consistency and repeatability of common manual tasks. These service designs then become the basis for pipeline management.

A sequenced service design includes the following:

- A hierarchy of service components (see "Service Components" on page 85). The hierarchy of service components defines the execution process. The hierarchy defines the connection rules, but the **Component Order** configured for each service component defines the order of execution that occurs when this design is provisioned.

- Resource offerings, which can be added to service components to enable the added resource offerings to be provisioned along with the service components.

- Lifecycle actions, which are needed to provision and de-provision service components. See "Lifecycle Actions for Service Components" on page 94.

- Properties, which allow you to configure user-defined properties for service components. See "Service Component Properties" on page 123

- Property mappings, which allow you to configure the property of a component to obtain its value from the property of another component. See "Create a Property Mapping" on page 120.

> **Note:** All features related to **Subscriber Options** and **Marketplace Portal** are available only with the CSA license applied.

To navigate to the sequenced design area

1. In the Management Console initial dashboard view, click the **Designs** sidebar menu item.

2. Click the **Sequenced** sidebar menu item, which takes you to the landing page for sequenced designs, allowing access to the **Designer**, **Components**, and **Resource Offerings** sidebar menu items.

To design a sequenced service

Follow these high-level steps to design a sequenced service:

1. Based on the needs of your organization, create a plan for resource providers, components, lifecycle actions, and other design elements before you begin implementing your design.

2. Make sure the resource providers and resource offerings (with lifecycle actions, as required) that are needed to deploy the service have been set up. See Providers .

3. Create component types, as needed (see "Create a Component Type" on page 181) or use out-of-the-box component types (see Components (Sequenced Designs)).

4. Create the service design. See "Add a Service Design" on page 70.

5. Use the **Designer** to create a hierarchy of service components. See "Sequenced Designer" on page 82 and "Create a Service Component" on page 86.

6. Associate resource offerings with service components, as needed. See "Associate Resource Offerings with Service Components" on page 105.

7. Create lifecycle actions for service components, as needed. See "Lifecycle Actions for Service Components" on page 94.

8. Create custom properties for service components, as needed. See "Service Component Properties" on page 123.

9. Map properties for service components, as needed. See "Create a Property Mapping" on page 120.

10. Complete the service design (blueprint) and publish the design. See "Add a Service Design" on page 70.

View All Service Designs and Versions

For more information about sequenced designs, see "Sequenced Designs" on page 60.

To view service designs and versions

This Sequenced Designs view displays a list of all service design tags, a list of all service designs, and up to five version numbers for each service design. View a list of service design tags in the left pane of the **All Designs** area. Click a tag to view the list of service designs and versions associated with the selected tag.

Use the following icons and features to navigate and perform tasks in the Designs area:

Item	Description
Search box	Enter text to filter the results based on a keyword search of display name, description, and version.
⚙	Click the gear icon to select any of the following actions: • **Create Design** - Add a new design. For more information about configuration options, see "Add a Service Design" on page 70. • **Import Design** - Import a service design. For more information, see "Import and Export a Service Design" on page 75. • **Manage Design Tags** - Manage tags. For more information, see "Manage Tags" on page 128. • **Refresh** - Refresh the data in this view.
Design Name	Click the name of a design to navigate to the **Overview** tab for the design. For more information about configuration options, see "View Service Design and View Service Design Version" on page 64.
New	Click **New** to create a new version of a particular service design. For more information about configuration options, see the "Create New Version" task in "View Service Design and View Service Design Version" on page 64.
Version Number	View the names of the latest five versions of the design sorted by creation date. These version names are links to the **Overview** tab of the Service Design Versions page. Click the version number of a design to display details about that version of the service design and see the layout of the design. If there are more than five versions, click the **All** link to open the **Versions** tab of the service design page, which displays a complete list of versions sorted by creation date. If there are no existing versions, only the **New** link displays in the Designs area. For more information about configuration options, see "View Service Design and View Service

Item	Description
	Design Version" on page 64. For more information about design versions, see "Design Versions" on page 78.

View Service Design and View Service Design Version

Design Tasks

Design Version Tasks

Concepts

The service design page displays details about a single specific service design. The **Overview** tab provides a summary view of the design, and the **Versions** tab provides a summary view of the design versions.

The service design version page displays details about a single specific service design version. The **Overview** tab provides a preview of the design.

Design Tasks

View Design Overview

View Versions

Create New Version

Edit a Design

Delete a Design

View Design Overview

In the **Overview** tab of the service design page, the following information about the design is displayed:

Item	Description
Display Name	Design name.
Description	Design description.
Tags	Tags associated with this design.

View Versions

In the **Versions** tab of the service design page, the following information about each version of the design is displayed:

Item	Description
Version	Design version.
Published label	Appears if the design has been published.
Upgradable label	Appears if the design has been published and can be upgraded to another version.
Description	Design description.
Creation date	Date and time the design was created.

Create New Version

On the service design page, click the gear icon and select **Create New Version** to create a new version of the design. If Based on Version is set to a published design, the new design can be made upgradable from the selected version. For descriptions of the specific properties, see the topic "Design Versions" on page 78.

Edit a Design

Click the gear icon and select **Edit** to modify the display name, description, image, or tags of the design. For descriptions of the specific properties, see the topic "Add a Service Design" on page 70.

Delete a Design

Click the gear icon and select **Delete** to delete the design. A design can only be deleted if no versions of it exist.

When a sequenced design is deleted, all the packages associated with it will be deleted as well.

Design Version Tasks

View Design Version Overview

Edit a Design Version

Publish/Unpublish a Design

Deploy a Design Version

"Cancel and Delete a Deployed Design Version " on page 68

Save As

Create New Version

Export a Version

Delete a Version

The service design versions page displays:

- **Design Name** and **Version** in the upper left.

- **Upgradable** label in the upper right. If visible, the design is available for upgrade to the listed versions in the **Upgradable To** section of the Overview tab.

View Design Version Overview

In the **Overview** tab of the service design versions page, the following information about the design version is displayed:

Item	Description
This design is part of an upgrade path and has restricted editability and actions.	This message appears when the design is part of an upgrade path. Certain functions are restricted on a design if it is part of an upgrade path.
Display Name	Design name.
Description	Design description.
Preview	Image of the design hierarchy. Clicking on the image opens a larger image in a separate dialog. Any changes to the design made in the dialog are not saved and do not affect the design as it appears in the sequenced designer.
Version	Design version.
URL	An optional URL that can contain more information about this design.
Published	Indicates if the service design is published (Yes) or not published (No).
Upgradable From	A link to the version of a service design that this design is based on and that can be upgraded to the current design version.
Upgradable To	A link to the version(s) of a service design to which this design can be upgraded.

Edit a Design Version

After saving a version, the following fields can be edited:

- **Version**

- **Description**

- **URL**

Publish/Unpublish a Design

Publish the currently displayed version of the design so it can be used to create a service offering. A published design cannot be modified. However, you can use **Save As** or **Create New Version** to create editable designs based on the published design. A published design is upgradable.

The Published status on the **Overview** tab is **No** until you click **Publish**.

After publishing, the **Published** status on the **Overview** tab displays **Yes** and the **Published** label appears in the upper right of the window. The **Published** label is also displayed in the **Versions** tab of the service design page.

You can unpublish the currently displayed version of the design so that service offerings cannot be created from it. If any service offerings have been created from the design or if the design is upgradable, it cannot be unpublished.

> **Note:** This option is available only when the CSA license is applied.

Deploy a Design Version

You can deploy the currently displayed version of the design.

Note: Before deploying, ensure that the design version contains a component layout. You cannot deploy an empty design version. You can create the component layout using the **Designer** tab.

To deploy a design version:

1. In the Codar console, click the **Designs** sidebar menu item and select **Sequenced** to display the **Sequenced** page.

2. Click **Designer** to display the **Sequenced Designs** page.

3. Select the sequenced design you need to deploy.

4. Select the version of the sequenced design you need to deploy.

5. Click the **Deployments** tab.

6. Click the **Deploy** button.

7. In the **Deploy a New Instance** dialog box, fill in the following fields:

Field	Description
Deployment Name	Name of the deployment.
Description	Description of the design version you are deploying.
Resource Environment Filer	Optionally restrict the provider selection to only those providers located in the selected environment.

8. Click **Next**.

9. Select a component from the design canvas and optionally modify the property values that will be used during provisioning.

10. Click the **Deploy** button.

After a sequenced design version has been deployed, an entry is made on the **Deployments** tab for the design. This tab lists all deployments, their status, date of creation/modification of deployment, and the user who created the deployment.

You can click the deployment instance to view status of the deployment. You can view the following information:

- Overview

- Events

- Topology

- Providers

Cancel and Delete a Deployed Design Version

You can cancel (undeploy) and delete deployed versions of sequenced designs. A deployed design version can be deleted only after canceling the deployment.

To cancel a deployment

1. Select the deployment instance to be canceled.

2. Click the gear icon on the right side and select **Cancel Deployment**.

3. Click 'Yes' in the confirmation dialog.

To delete a deployment

1. Select the deployment instance to be deleted.

   Note: You can delete only those deployments that you own.

2. Click the gear icon on the right side and select **Delete Deployment**.

3. Click 'Yes' in the confirmation dialog.

Save As

Using **Save As** creates a new initial starting version of a new design:

Create New Version

On the service design page, click the gear icon and select **Create New Version** to create a new version of the design. If Based on Version is set to a published design, the new design can be made upgradable from the selected version. For descriptions of the specific properties, see the topic "Design Versions" on page 78.

Export a Version

See "Import and Export a Service Design" on page 75.

Delete a Version

You cannot delete a version if it has an offering associated with it. Deleting an upgradable or upgraded design also deletes the upgrade path associated with the design.

Add a Service Design

For more information about sequenced designs, see "Sequenced Designs" on page 60.

To add a service design

1. In the **All Designs** area, click the gear icon and select **Create Design**.

2. Enter information described in the following table, and then click **Create.**

Item	Description
Display Name	The name provided for the design is shared by all versions of the design.
Description	The description you provide for the service design. Individual versions of a design can contain their own descriptions.
Initial Version	Provide the number or name for the initial version of the design.
Image	The image that displays for the service design. The image selected for the design is shared by all versions of the design. Click **Change Image**. Choose the image you want, and click **Select**. Click **Upload** to add your own image. Supported file extensions include `.jpg`, `.jpeg`, `.gif`, and `.png`. The recommended image size is 256 by 256 pixels, and the image will be scaled to the appropriate size. The images are stored in the `%CSA_HOME%\jboss-as\standalone\deployments\csa.war\images\library` folder of the Codar server.
Tags	Click **Select Tags** to choose from a list of tags that you can define to provide a structure for organizing and grouping the service designs. For more information, see "Manage Tags" on page 128. To remove a tag, click the trashcan icon.

3. Once the design has been added successfully, select the **Designer** tab to add service components to the design.

For more information about service components, see "Service Components" on page 85.

Edit Service Design Properties

For more information about sequenced designs, see "Sequenced Designs" on page 60.

To edit properties of a service design

1. Select the design you want to edit.

2. Click the gear icon and select **Edit**.

3. In the Edit Design dialog, modify properties of the service design. For descriptions of the specific properties, see the topic "Add a Service Design" on the previous page.

4. Click **Save**.

Copy a Service Design

For more information about sequenced designs, see "Sequenced Designs" on page 60.

To copy a service design

To copy a service design, you must select a specific service design **Version** to copy. For more information about versions, see "Design Versions" on page 78.

1. Select the design version you want to copy.

2. Click the gear icon and select **Save As**.

3. In the Save As dialog, enter a new display name and version.

4. Click **Save**. The new service design is displayed.

All parts of a service design (such as the service component hierarchy and properties) are copied. The copied service design appears in the **All Designs** area. After you copy a service design, any changes you make to the original or to the copy will not affect the other.

Delete a Service Design

For more information about sequenced designs, see "Sequenced Designs" on page 60.

A service design cannot be deleted if any deployments exist. In addition, a service design cannot be deleted until all versions of the design have been deleted. If a service design is deleted and it is part of an upgrade path, then the upgrade path is also deleted. When a service design is deleted, all the packages associated with it will be deleted as well.

To delete a service design

1. Select the design you want to delete.

2. Click the gear icon and select **Delete**.

3. Click **Yes** to confirm the deletion.

Tag a sequenced design for Release Automation

Tagging allows users to filter the designs available for Release Automation. Only the sequenced designs that are tagged for Release Automation will be available in release pipeline view.

To tag a sequenced design as an application

1. From the Codar Dashboard, navigate to **Designs** >> **Sequenced** >> **Designer**.

2. In the **All Designs** area, click the design you want to tag for Release Automation.

3. On the 'Overview' tab of the design, click the gear icon and select 'Manage Release Automation'.

4. In the 'Manage Release Automation' dialog box, select the 'Application design' checkbox.

5. Click **Save**.

A sequenced design tagged as an application is indicated by appending the word 'Application' next to the design in the **All Designs** area.

To untag a sequenced design

In the 'Manage Release Automation' dialog box, uncheck the 'Application design' checkbox.

> **Note:** If you created packages for a sequenced design and you cannot see the same in release pipeline, ensure that you have tagged the respective sequenced design as an application. Unless a design is tagged, packages belonging to the design will be hidden from users.

Import and Export a Service Design

For more information about sequenced designs, see "Sequenced Designs" on page 60.

Prerequisites

Before you import a service design archive, complete the following prerequisites:

1. Import all process definitions that are referenced by the service design, including any referenced by resource offerings that are part of the service design. The process definitions must be imported to the Codar installation in which the service design will be imported. During import, process definitions are resolved by **name**. For process definitions that are associated with the Operations Orchestration process engine, the **name** corresponds to the full path to the Operations Orchestration flow (for example, `/Library/CSA/3.2/Providers/vCenter/vCenter Clone Server/Actions/vCenter Simple Compute - Deploy`). Make sure that these flows have identical signatures and identical paths as the flows on the system from which the service design was exported. For instructions about how to import OO flows, see the section "Import Operations Orchestration Flows" in the *Configuration Guide*.

2. If the service design you want to import has a dependency on any custom component types, these custom component types must be imported before the service design can be imported.

3. If the imported service design is part of an upgrade path, all service designs upgradable to the service design represented by the content archive must be present on the target system.

The archive (`.zip`) file can only reference files or content contained within the `.zip` file itself, or files that are already contained in the `csa.war` file. By default, all images in the archive files must end in one of the following suffix values. For information on adding additional suffix values, see the *Configuration Guide*.

`jpg|jpeg|jpe|jfif|svg|tif|tiff|ras|cmx|ico|pnm|pbm|pgm|ppm|rgb|xbm|xpm|xwd|png|gif| bmp|cod|ief`

Importing and Exporting

When the Option field is set to Import:

The import process imports archives of service designs and their supported artifacts. Supported artifacts for service designs include associated resource offerings. Service designs with the same internal name are considered to be functionally equivalent and are not imported.

> **Note:** If the service design you want to import is part of an upgrade path and is also upgradable to another service design, it will not be imported.

When the Option field is set to Update:

This is an import-by-update option. During the update process, identical artifacts that exist on the target system are updated (overwritten) with the changes from the archive. Artifacts are created if they do not exist on the target system.

> **Note:** If the service design you want to import-by-update is part of an upgrade path and is also upgradable to another service design, it will not be imported.

When the Option field is set to Update and Preserve Originals is selected:

This is an update-by-preserve-original option that applies only to service offerings that do not have an upgrade path. This action imports all artifacts present in an archive, whether they exist on the target system or not. During this operation, if there is a service design with the same internal name in the system, then the name is modified internally, and the description and version of that service design are updated with "Superseded on" and the date. The internal name, display name, description, and version of the artifact being imported remain intact. Other artifacts referenced by the service design are handled similarly, with superseded items identified by "Superseded on" and the date.

> **Note:** When the Option field is set to Update and Preserve Originals is checked, you cannot import an archive if the service design is part of an upgrade path.

Content archive for service designs

Exporting a service design creates a content archive (`.zip`) file. The content archive contains XML documents for the service design you are exporting, as well as associated artifacts, icons for customizing the artifacts, and the Manifest XML document, which contains meta-information about the archive files.

To import a service design version archive

If a service design has active service offerings associated with it, the subscriber options for the service design being imported must match that of the service design on the target system; otherwise, the import of the service offerings does not succeed.

> **Notes:** When you import a design, any tags you have selected in the designs area do not affect the imported design. The imported design will contain the tags that were included when the design was exported, and new tags will be created on the system, as necessary, to match what was exported.

1. In the Sequenced Designs view where All Designs displays, click the gear icon and select **Import Design**.

2. In the Import Design dialog, select or specify the **Archive File** (.zip file) that contains the service design you want to import. Archive filenames for service designs begin with SERVICE_ DESIGN_.

3. Select an **Option**:
   a. **Import** - Imports a new service design and its associated resource offerings from the selected archive file, if it does not already exist. This action does not update existing service designs. You cannot import a service design if it has the same internal name as the existing service design.

   b. **Update** - Imports an existing service design and its associated resource offerings from the selected archive file. If the service design already exists, this action updates (overwrites) existing service design versions. If the service design does not exist, this action will add it to CSA. Check **Preserve Originals** to create a backup copy of the existing service design. The backup copy will be saved with "Superseded" and the date appended to the display name of the service design.

4. Click **Preview** to see a report of prospective results for the import process, including information about the artifacts and their status.

5. Click **Import** and wait for the confirmation message. Click **View Detailed Report** to see a summary and details of the import process, including information about the artifacts and their status.

6. Click **Close**. The new version is listed in the **All Designs** area.

Exporting

1. Select the design version you want to export.

2. Click the gear icon and select **Export**.

When you export a service design archive, the service design and all referenced resource offerings are packaged in an archive file named:

SERVICE_DESIGN_<service_design_display_name>_<service_design_id>.zip

For more information about importing and exporting artifacts, see the *Cloud Service Automation Content Archive Tool* document.

Design Versions

For more information about sequenced designs, see "Sequenced Designs" on page 60.

Concepts

Versioning duplicates an existing design and then allows you to configure and edit the new version. An initial version is automatically created when you create a new design, and you can create as many versions of the design as you wish. While a new version inherits the display name and icon from the original design, you provide a version number (or string), description, and URL that are unique to a particular service design version. Once a version is created, you can use the **Designer** to create a hierarchy of service components. See "Sequenced Designer" on page 82 for more information.

Any one or all versions of a design can be published. A design cannot be edited after it is published.

Tasks

On the service design page, click the gear icon and select **Create New Version** to create a new version of the design. If Based on Version is set to a published design, the new design can be made upgradable from the selected version.

Provide the information in the following table and click **Save**.

Item	Description
Based on Version	The version of the service design from which this new version is copied.
Version	Version of the new service design to be created.
Upgradable from *<Based on Version>*	If the base version of the design (the version from which the new version is copied) has been published, select if this new design is upgradable from the base version. If the base version of the design has not been published, this property is not available.
Description	Description of the new version of the design.

Publish a design version

Publishing a design version makes it available as an offering to service consumers. You must have a CSA license installed before you can publish a design.

A complete design with an active package in the Production stage contains package-specific properties as part of the design and can be published.

For topology designs, a partial design with an active package in the Production stage contains package-specific properties as part of the design, but it cannot be published until a final composed design is created by deploying the production package.

Publishing a partial design is different depending on which licenses you have installed:

- A Codar application design that has been advanced to the Production stage is deployed on a production infrastructure, and then the composed production design is made visible on successful production deployment. The design can then be published to service consumers.
- A design that is not a Codar application design must be saved as a composed design from the Test tab. The design can then be published to service consumers.

Notes:

- A sequenced design can be published only if it has no packages or if the package is in the last stage of Pipeline Management, with an active deployment instance.
- If a topology design has been tagged with an application, it cannot be published unless it is untagged.

The following figure illustrates when a sequenced design can be published based on the license used:
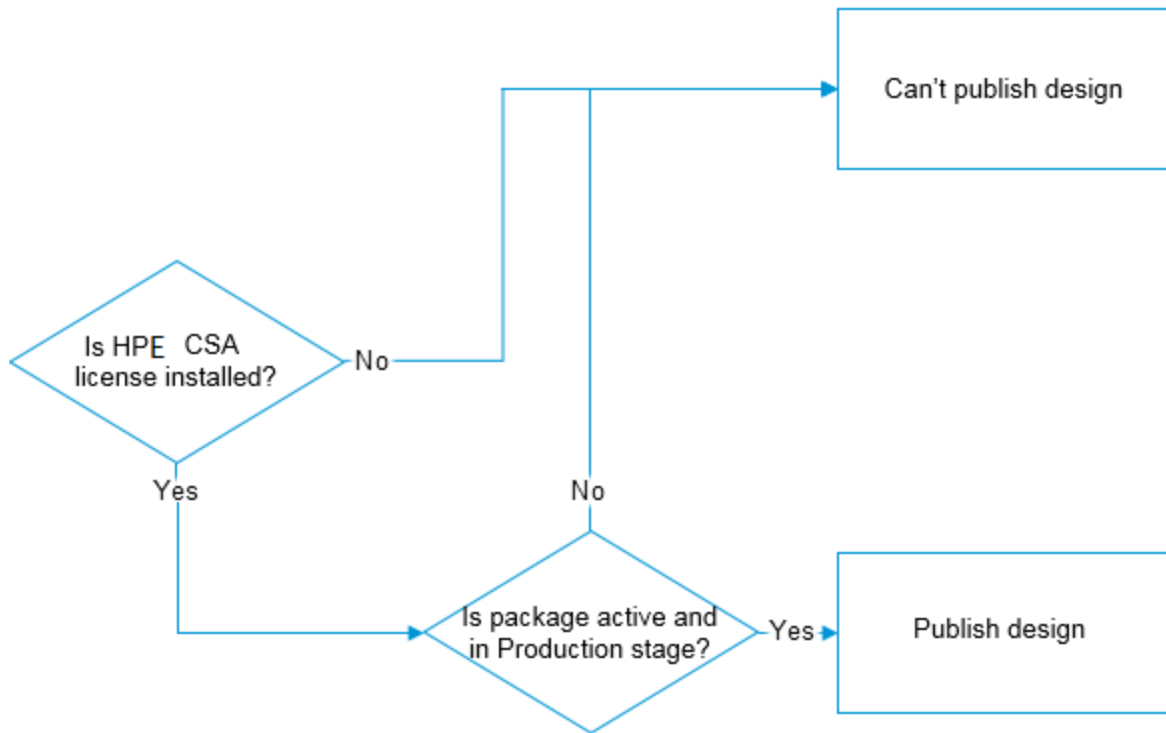
The following figure illustrates when a topology design can be published based on the license used:
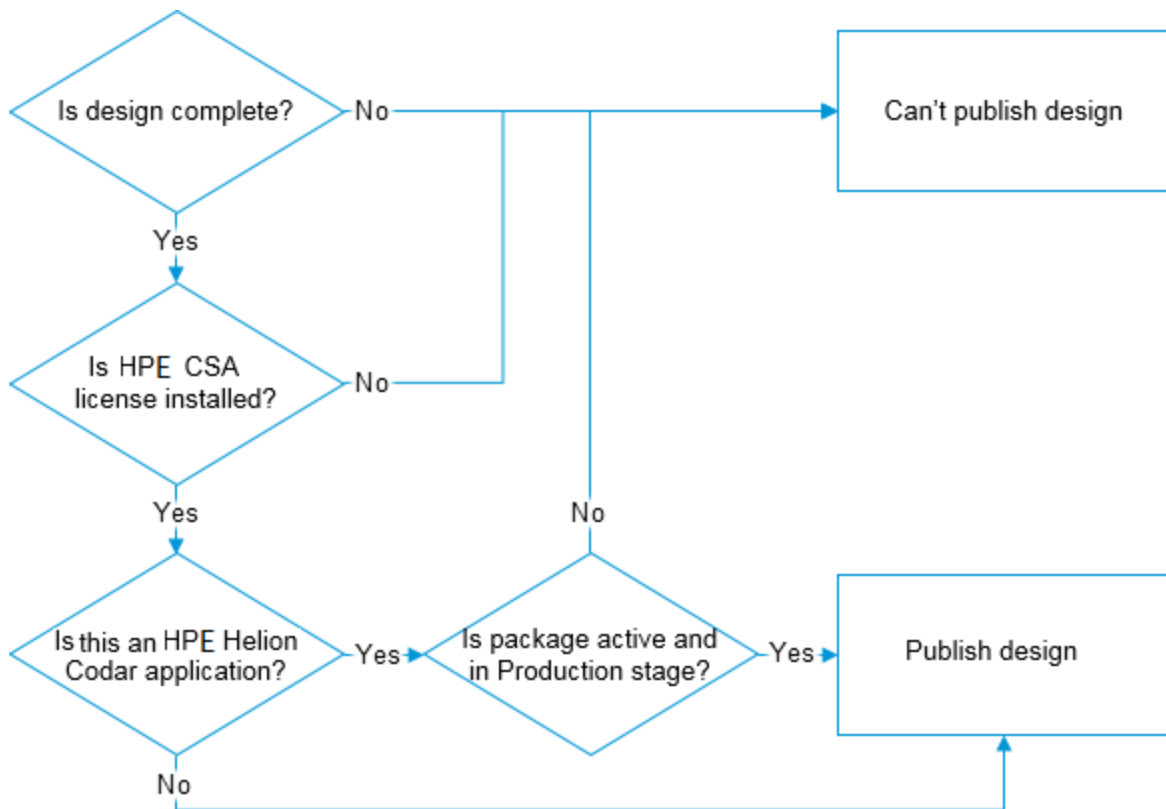
To publish a design

1. In the Codar Management Console, click Designs.

2. Select the type of design you want to publish (Sequenced or Topology).

3. Click **Designer**.

4. Click the version of the design you want to publish.

5. Click the gear icon on the right hand side and select**Publish**.

Sequenced Designer

For more information about sequenced designs, see "Sequenced Designs" on page 60.

The **Designer** allows you to design your sequenced services by creating a hierarchy of service components. A service component is an element of a service design. A service component has a component type that constrains its allowed children and assignable resource offerings.

An example hierarchy of service components, as shown below, has a Service Composite that contains an Infrastructure Service, which contains a Server Group and a Virtual Network. Also, the Server Group contains a Server (marked as a Pattern), while the Virtual Network contains a Policy Enforcement Point.
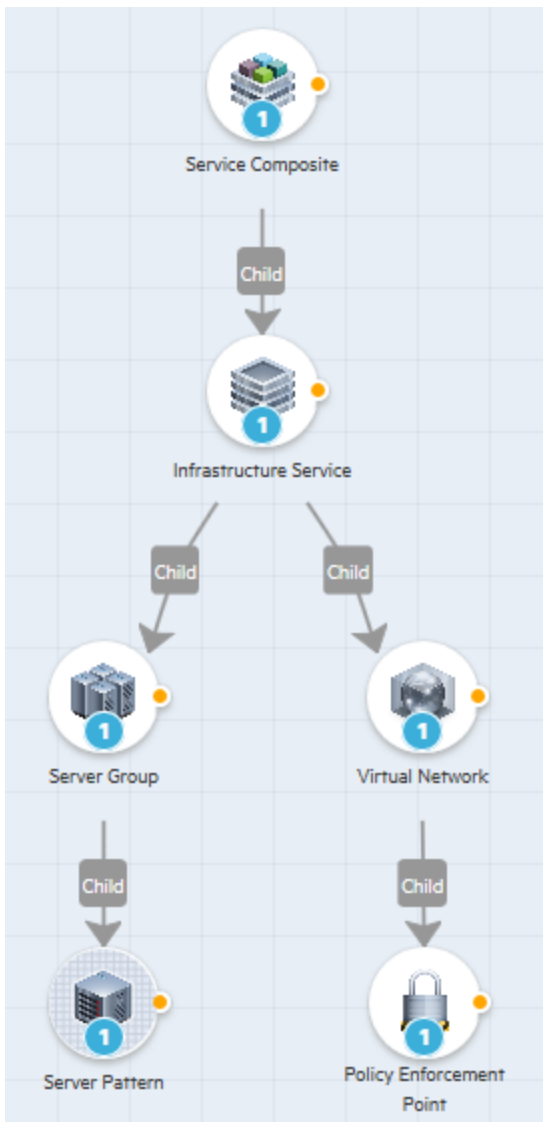
To navigate to the sequenced designer

1. In the Codar dashboard, click **Designs**, and then click **Sequenced**.

2. Click the **Designer** sidebar menu item. The **Sequenced Designs** page appears.

3. Select a service design version. If no versions exist, click **New**. For more information about versions, see "Design Versions" on page 78.

4. Click the **Designer** tab. The design canvas appears, with a list of palettes and list of components in the left pane. You can add components from the left pane to the design canvas to create your design, as described in "Create a Service Component" on page 86.

To use the designer controls

The **Designer** provides the following controls to help you create and view your service design.

Using zoom and reload/reset control options:

Icon	Description
🔍 | 🔍	Click to zoom the display out or in.
🔗	Click to reload the service design and reset the service components to their default positions.

Using service component control options:

Icon	Description
+ ✎ 📦 🗑	Click the orange dot on a service component on the canvas to display this toolbar. Use this toolbar to create a new child service component, edit a component, select a component template, or delete a component.

Service Design Validation

For more information about sequenced designs, see "Sequenced Designs" on page 60.

Validation errors are indicated on the design canvas with an error icon (⚠) on the component and on the designer control toolbar. Validation warnings are indicated on the design canvas with a warning icon (⚠) on the component and on the designer control toolbar. If a component has both warnings and errors, an error icon is displayed. Hover over the icon on the canvas to view details about the error or warning.

A service design with validation errors is not publishable.

A validation error indicates that your service design has one or more of the following problems:

- The service design has a component that does not have an established parent-child relationship to another component in the design.

- The service design has a pattern component that does not have a parent component.

- The service design has a pattern component whose parent component has a Clone Pattern action that does not have a value in the "Name of the Property for Service Component Count" property field or has a value that does not point to the name of a property on the component.

A validation warning indicates that your service design has one or more of the following problems:

- The service design has a pattern component whose parent component does not have the Clone Pattern action.

- The service design has one or more associated resource offerings that do not have enabled resource providers. If this service design problem is not corrected, the design cannot be successfully provisioned.

Service Components

For more information about sequenced designs, see "Sequenced Designs" on page 60.

A Service Component is an element of a service design. A Service Component has a component type that constrains its allowed children and category of resource offerings (for more information, see ""Create a Service Component" on the next page" .).

See the following related topics:

- "Create a Service Component" on the next page

- "Edit a Service Component" on page 88

- "Create a Property Mapping" on page 120

- "Change the Service Component Template" on page 116

- "Delete a Relationship between Service Components" on page 122

- "Delete a Service Component" on page 119

- "Service Component Properties" on page 123

Create a Service Component

For more information about sequenced designs, see "Sequenced Designs" on page 60.

To create a service component

1. Complete the steps in "To navigate to the sequenced designer " on page 83 and select the **Designer** tab.

2. Use the list in the left pane to select a component palette. A component palette contains a group of component types from which you can choose when creating service components in a service design.

3. The component types available for this palette appear. Use the search box to filter the component type list by display name and description. Select a component type.

4. When you select a component type, the templates available for that component type appear.
   - A search box appears for component types with five or more templates. Use the search box to filter the component template list by display name and description.

   - A component template is a specialized version of a component type and is used to simplify service design creation. Component templates can include customized property values and lifecycle actions that provide initial settings for the service design. When a template is added to a design, any default properties, bindings, and lifecycle actions are automatically populated in that service component within the design. This list always includes a 'Blank Template,' which uses the settings as configured for the selected component type.

5. Select the template of the component type you want, and add it to the design canvas on the right using any of the following methods. You can add parent or child components using these methods, and you can add as many components as you need for your design:
   - Drag and drop the template to the canvas.

   - Doubleclick on a template to add it to the canvas.

   - Click the orange dot on a service component on the canvas to bring up the designer controls toolbar (see "To use the designer controls" on page 83). Click the **Add** icon ＋ to add a parent or child service component.

6. To establish parent-child relationships between component types on the canvas, click the orange dot on a service component on the canvas, and draw a line to the parent or child component type.

   If a relationship is defined incorrectly, component type constraints will not allow you to create the relationship. If a component does not have an established parent-child relationship, a validation error is displayed and the design cannot be published.

7. Click the Automatic Layout icon 🏛 to organize and display your design in a graphical format.

For more information about component palettes, types, and constraints, see Components (Sequenced Designs).

Edit a Service Component

Double-click a component, click the **Edit** icon , or click the gear icon and select **Edit Component** to bring up the **Edit Component** dialog.

The Edit Component dialog allows you to configure attributes such as Processing Order and Pattern, configure component properties, configure lifecycle actions, configure user operations, and configure resource offerings.

Use the functional area tabs to edit a service component:

- ""Overview Tab for Service Components" on the next page"

- ""Properties Tab for Service Components" on page 91"

- ""Lifecycle Actions for Service Components" on page 94"

- ""User Operations for Service Components" on page 102"

- ""Associate Resource Offerings with Service Components" on page 105"

Overview Tab for Service Components

For more information about service components, see Components (Sequenced Designs).

Concepts

The Edit Component Overview tab allows you to modify attributes of a service component, including its Display Name, Description, Processing Order, and Consumer Visible and Pattern settings. For more information about these properties, see "Edit a Service Component" on the previous page and "Create a Component Type" on page 181.

Tasks

- **Edit** - If a field is editable, enter data or select/clear a check box.

  For a published design, you cannot edit the following attributes:

  ○ **Display Name** - The name you provide for the service component.

  ○ **Description** - The description you provide for the service component.

  ○ **Processing Order** (default value is 1, minimum value is 1, maximum value is 99) - Select a number to specify the deployment processing order for this service component relative to its sibling service components (that is, service components who share the same parent). Service components are processed in ascending order during service deployment and in descending order during undeployment.

  ○ **Image** - An image appears for the component. Click **Change Image**, as desired. The **Select Image** dialog appears, allowing you to choose from a variety of available images or click **Upload Image** to add your own image.

  ○ **Consumer Visible** (the default is false) - Select Visible to specify that this service component is visible in the . De-select it to specify that the service component will not be visible. Service component properties and lifecycle actions that have been configured to be visible in the will only be displayed in the if the associated service component is itself configured to be visible. The visibility of a service component does not affect visibility of its child service components, and it is not necessary for a parent of a service component to be visible for the child component to be visible in the .

- **Pattern** (the default is false) - Select this box to mark the service component as a pattern. This indicates that the service component will not be automatically processed by the lifecycle engine. When you create a pattern service component, you must also create a Clone Pattern action in the parent service component. For example, the out-of-the-box component type, Server, is typically marked as a pattern, and its parent service component, Server Group, includes the Clone Pattern action. At subscription time, the Clone Pattern action creates one or more service component instances from this service component pattern. A pattern component can have direct children, which are not patterns. When a pattern component contains a direct child component, the Clone Pattern action creates one or more instances of the pattern including the child component.

  - A validation error is displayed in the designer when a parent-child relationship is established between a pattern service component and its parent component if the Clone Pattern action is not created in the parent service component.

  - A validation warning is displayed when the service design has a pattern component whose parent component does not have the Clone Pattern action.

  - A validation error is displayed when the service design has a pattern component whose parent component has a Clone Pattern action that does not have a value in the "Name of the Property for Service Component Count" property field.

  - A validation error is displayed when the service design has a pattern component whose parent component has a Clone Pattern action that does not have a value in the "Name of the Property for Service Component Count" property field or has a value that does not point to the name of a property on the component.

  - You can identify a service component that is marked as a pattern because its icon appears with a grid-like background, as shown below.
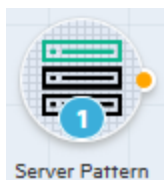
  Server Pattern

  If a design that is in an upgrade path, is upgradable from a previous version of the design, and the component existed in that prior version, you cannot edit the following attributes:

  - Processing Order

  - Pattern

- **Save** - Click **Done** to save any changes and exit the dialog.

Properties Tab for Service Components

For more information about modifying the attributes of a service component, including its Display Name, Description, Processing Order, and Consumer Visible and Pattern settings, see "Overview Tab for Service Components" on page 89.

Concepts

The following information is displayed:

Item	Description
Type	Determines the kind of value that can be assigned to the property. The available types are: • Boolean • Integer • List • String You cannot change the type once the property has been created.
Display Name	The label that will be displayed for the property. Hover the cursor over a display name to show the unique name (**Name**) and description (if supplied) of the property.
Value	The value of the property. What is entered for the value depends on the property type: • **Boolean** - A property whose value is true or false. • **Integer** - A property whose value is a positive or negative whole number (or zero). The **Resource Type and Unit for a Measurable Property**, if configured, is displayed in parentheses (see "Create Component Type Properties" on page 187 for more information). • **List** - A property whose value is a list of string values. • **String** - A property whose value is a sequence of characters. If the property has a property mapping, the value displayed is the component and property from which the value is mapped. Property mappings are configured from the design canvas. See "Create a Property Mapping" on page 120 for more information. **Note:** If a property has a target binding, by default (until runtime) the value displayed is the value that was configured when the property was created. If the property is mapped to a token (String properties only), the value displayed is the name of the token (a token is a system value that is automatically resolved internally when the property is read).

Item	Description
	If the property is configured as **Confidential Data** (String properties only), the value is masked (no encryption of the value is performed).
References	The total number of component properties using this property as the source (the value of this property is mapped *to* another component property) and target bindings. A component property that has a property mapping (its value is mapped *from* another property) is not considered a reference. Click the reference to display the component properties to which this property is mapped and the subscriber option properties from which this property has a target binding.
Attributes	The attributes configured for the property such as: • **Inherited** - The property is inherited from a component type. • **Consumer Visible** - The property will be made visible in the Marketplace Portal. **Note:** Marketplace Portal is available only when the CSA license is applied.
Actions	The gear icon, when selected, allowing you to edit or delete the property (if the action is permitted on the property). See the descriptions of these actions in the Tasks section below.

Tasks

- **Navigate to** (the Properties tab) -Select a service design version. Click the **Designer** tab. Select the service component on which you want to complete the task. In the right pane, click the gear icon, select **Edit Component**, and select the **Properties** tab.

- **Create** - Navigate to the Properties tab. Click **Create**. Enter the information described in the table above (see also "Create Component Type Properties" on page 187) and click **Create**.

  A service component property cannot be created if the service design is published.

  > **Note:** Property mappings are configured from the design canvas. See "Create a Property Mapping" on page 120 for more information.

- **Edit** - Navigate to the Properties tab. Click the gear icon next to the component property to be modified and select **Edit**.

  A service component property cannot be edited if the service design is published.

The Name of a property cannot be edited if the design is in an upgrade path, is upgradable from a previous version of the design, and the component existed in that prior version.

- **Delete** - Navigate to the Properties tab. Click the gear icon next to the component property to be deleted, select **Delete**, and click **Yes** in the confirmation dialog.

A service component property cannot be deleted if the design is in an upgrade path, is upgradable from a previous version of the design, and the component property existed in that prior version. Component properties that are newly added to the current version of the design can be deleted.

If a property has references, Codar removes these references before deleting the property.

Lifecycle Actions for Service Components

For more information about service components, see "Service Components" on page 85.

Concepts

What is a service component lifecycle?

A service component lifecycle is a collection of actions defined for a service component. The Lifecycle area allows you to specify the actions that are needed to provision, update, and deprovision service components. Use the action selection wizard to add a lifecycle action to a service component from one of the available process engines.

What is a lifecycle action?

A lifecycle action is a function that is either run automatically at a specified lifecycle stage or phase, or that is exposed to the user. Most lifecycle actions correspond to OO flows, which contain the logic for executing the function. Codar also includes actions that run internal to Codar. Many Codar internal actions relate to provider and pool selection and resource accounting. Actions include input parameters that provide configuration information to the function.

What are lifecycle stages?

A lifecycle stage represents a step within the Codar service lifecycle. There are three categories of lifecycle stages: Provisioning, Operational, and De-provisioning.

The Provisioning lifecycle stages are:

- **Initializing** - Initializing is the first stage that is processed during provisioning. Use the Initializing stage for actions that perform any type of initialization that is required before proceeding to the Reserving stage, such as actions that perform input validation or create change request records.

- **Reserving** - Reserving is the second stage that is processed during provisioning, after the Initializing stage. When a resource offering is associated to a service component, any configured provider selection and pool selection actions run during the Reserving stage. If resource accounting is enabled, actions to reserve resources from a pool run during the Reserving stage. In general, any resources needed during provisioning, such as storage or networking, can be reserved during the Reserving stage.

- **Deploying** - Deploying is the final stage that is processed during provisioning, after the Initializing and Reserving stages, and is the most common stage in which provisioning actions are defined. Define actions in the Initializing and Reserving stages to prepare the environment for provisioning.

Define actions in the Deploying stage to perform actual provisioning. For example, you can define actions that deploy a VM or an application in the Deploying stage.

The De-provisioning lifecycle stages are:

- **Un-deploying** - Un-deploying is the first stage that is processed during de-provisioning and is the most common stage in which de-provisioning actions are defined. Use the Un-deploying stage to undo or to finalize the actions that took place during the Deploying stage. For example, if actions in the Deploying stage deploy a VM, define actions in the Un-deploying stage to un-deploy the VM.

- **Un-reserving** - Un-reserving is the second stage that is processed during de-provisioning, after the Un-deploying stage. Use the Un-reserving stage to undo or to finalize the actions that took place during the Reserving stage. For example, if resource accounting is enabled, actions to return resources to the pool run during the Un-reserving stage.

- **Un-initializing** - Un-initializing is the final stage that is processed during de-provisioning, after the Un-deploying and Un-reserving stages. Use the Un-initializing stage to undo or to finalize the actions that took place during the Initializing stage. For example, if actions in the Initializing stage create a change request, define actions in the Un-initializing stage to finalize that change request.

What are lifecycle phases?

A lifecycle phase is a further refinement of a lifecycle stage. When defining a lifecycle action at a lifecycle stage, you can also specify the phase for the action. The phases are:

- **Before** - Actions configured in the Before phase on a service component run before all actions on child service components in a design and before all actions on any associated resource offerings. Actions configured in the Before phase on a resource offering run after all of the Before actions on the associated service component have completed.

- **During** - Actions configured in the During phase on a service component run after all of the actions have completed on child service components in a design but before any of the During actions run on any associated resource offerings. Actions configured in the During phase on a resource offering run after all of the During actions on the associated service component have completed.

- **After** - Actions configured in the After phase on a service component run after all actions have completed on child service components and after all of the During actions have completed on the service component and any associated resource offerings. Actions configured in the After phase on a resource offering run after all of the After actions on the associated service component have completed.

- **On Failure** - Actions configured in the On Failure phase on a service component or resource offering run after an action failure occurs in any of the Before, During, or After phases for that service component or resource offering.

Tasks

> **Note:** If a service design is published, its lifecycle actions are read-only.

- **Navigate to** (the Lifecycle tab) - In the left pane of the **All Designs** area, select the tag associated with the design you want to view. Select the service design and, in the **Designer** tab, select the service component on which you want to complete the task. In the right pane, click the gear icon, select **Edit Component**, and select the **Lifecycle** tab.

- **Add Action** - Click the **Add Action** or **Add More Actions** link in the stage and phase to which you want to add an action. Enter the information as described in "Lifecycle Action Selection Wizard" on the next page.

- **Edit Action** - Select the gear icon on the action you want to edit and select **Edit**. Update the information as described in "Lifecycle Action Selection Wizard" on the next page.

- **Delete Action**- Click the gear icon next to the action you want to delete and select **Delete**. You can also delete an action that is in a group.

- **Move Action**- Select the gear icon on the action or group of actions you want to move to another stage or phase and select **Move**. In the Move dialog, select the Target Stage and Target Phase from the pull-down menus and then click **Move**. You can also click and drag to move an action or group of actions. To move an action or group of actions using the keyboard, press the tab key to provide focus on the action or group and then press Ctrl and the up or down keys to move the action or group within a phase.

- **Group Actions** - Click to select multiple actions in a lifecycle phase and then click **Group Actions** to create a parallel execution group. Actions in a parallel execution group are executed at the same time.

- **Ungroup Actions** - Click the gear icon next to an action in a group and then select **Remove from Group**.

- **View Lifecycle Stages** - The Deploying Provisioning stage, the Modifying Operational stage the Un-deploying De-provisioning stage, and any stage that has an action, are displayed by default. Select a filter from the All Selected Stages pull-down menu to display only Provisioning Stages, Operational Stages, or De-provisioning stages. Click **Select Stages** to view additional stages.

- **View Lifecycle Phases** - The During and On Failure phases are displayed by default. Select **Show All Phases** to display the Before and After phases.

Lifecycle Action Selection Wizard

See also Lifecycle Action Selection Wizard (for resource offerings) or Lifecycle Action Selection Wizard (for service component templates).

Concepts

Use the action selection wizard to create or edit an action for the selected lifecycle phase.

To add Operations Orchestration flows, see the Import Operations Orchestration Flows section in the *Configuration Guide* for more information.

Tasks

- **Select Action** - Search for flows or actions by name (when searching for a flow, the folders searched in the Operations Orchestration library are determined by a property configured in the `csa.properties` file; see the "Action Selection Wizard" property description section in the Configuration Guide for more information) or select the process engine from which to select a flow or action. Then, locate and select the flow or action. See "Lifecycle Action Internal Actions" on page 99 for a list of internal actions. This task is only available when you create an action.

- **Configure Action** - Configure information about the action.

Item	Description
Process Engine	Read-only. The container of the flow or internal action.
Locator	Read-only. The location of the flow or internal action.
Display Name	The name you provide for the action.
Description	The description you provide for the action.
Execution Properties	Specify the following: ○ Fail on Error - If selected, provisioning or de-provisioning will stop if the action fails. By default, Fail on Error is not selected for actions created in the Un-deploying, Un-reserving, and Un-initializing stages. ○ Error on Timeout - If selected, provisioning or de-provisioning will stop if the time

Item	Description
	to execute the action exceeds the timeout value, configured below.
Timeout (seconds)	The amount of time to wait for the action to complete. Set this field to zero (0) if you do not want the action to time out. Note that this creates the potential of an action not finishing, which may cause the underlying subscription to remain in the Pending state.

- **Configure Parameters** - Configure the input parameters whose values are passed to the flow or internal action prior to invoking the action. Click the gear icon next to the parameter you want to configure and select **Edit Mapping**. From the Edit Parameter Mapping dialog, edit the Name, Display Name, Description, and Mapping Type of the parameter.

Select one of the following Mapping Types:

- Not Mapped - The parameter is not mapped to a value.

- Constant Value - Type a value for the parameter.

Lifecycle Action Internal Actions

For more information about lifecycles, see "Lifecycle Actions for Resource Offerings" on page 150.

The following internal actions are available with Codar:

Internal Action	Description	Applies To
Build Resource Provider and Pool List	Builds a candidate list of resource providers and associated resource pools that meet the following requirements: • Support the resource offering. • Have an **Availability** of **Enabled**. • The provider's resource pool has sufficient resource capacity. To determine this, you must consider all measurable properties as configured in the **Measurable Properties** tab for the resource offerings, as well as the optional **Multiplier Property Name** field. The resource pool must have enough resource capacity to support all the properties, which requires that each necessary resource type (for example: CPU, Memory, and Storage), based on the measurable properties, be configured on the pool either with a **Resource Availability** of **Unlimited** or **Available**. If **Available**, the difference between **Total Available To CSA** and **Current CSA Utilization** must be sufficient to support the measurable property requirements.	Resource Offering associated with a Service Component
Build Resource Provider List	Builds a candidate list of resource providers that meet the following requirements: • Support the resource offering. • Have an **Availability** of **Enabled**.	Resource Offering associated with a Service Component
Clone Pattern	Clones a service component that is marked as a **Pattern** into one or more non-pattern service components. The number of service components created is determined by the value of the property specified in **Name of the Property for Service Component Count**.	Service Component
Decrease Resource Utilization	Decreases the utilization of one or more resources in a resource pool by the values of the measurable properties configured on a resource binding. This action should be configured in the **Un-reserve** section of the **Resource Accounting** tab on a resource offering associated with a service component. **Note:** :The Decrease Resource Utilization action cannot be added to a component, component template, or resource offering. This action runs automatically during Un-reserving when Resource Accounting is enabled for a resource offering associated with a service component.	Resource Offering associated with a Service Component

Internal Action	Description	Applies To
Increase Resource Utilization	Increases the utilization of one or more resources in a resource pool by the values of the measurable properties configured on a resource binding. This action should be configured in the **Reserve** section of the **Resource Accounting** tab on a resource offering associated with a service component.. **Note:** :The Increase Resource Utilization action cannot be added to a component, component template, or resource offering. This action runs automatically during Reserving when Resource Accounting is enabled for a resource offering associated with a service component.	Resource Offering associated with a Service Component
Log Messages	Writes the user-specified Boolean Input, Integer Input, and String Input property values to the `csa.log` file. You can include this as an action on a Service Component, Resource Offering associated with a service component, or Resource Offering for use in troubleshooting.	Resource Offering Resource Offering associated with a Service Component Service Component
Select Resource Provider	Selects a resource provider from the candidate list that was built by the **Build Resource Provider List** action. The selected resource provider will be available to resource offering actions in the token `RSC_PROVIDER_ID`. The selected provider will, optionally, be written to a property on the associated service component if the `Provider Property Name` input to the action is provided.	Resource Offering associated with a Service Component
Select Resource Provider and Pool	Selects a resource pool and provider from the candidate list that was built by the **Build Resource Provider and Pool List** action. The selected resource provider and pool will be available to resource offering actions in the token `RSC_PROVIDER_ID` and `RSC_POOL_ID`, respectively. The selected pool will, optionally, be written to a property on the associated service component if the `Pool Property Name` input to the action is provided.	Resource Offering associated with a Service Component
Select Resource Provider and Pool from Parent	Selects the resource pool and provider already chosen by a service component's parent service component, as identified by the `Parent Component ID` and `Pool Property Name` properties. The selected resource provider and pool will be available to resource offering actions in the token `RSC_PROVIDER_ID` and `RSC_POOL_ID` respectively. The selected pool will also be written to a `Pool Property Name` property on the associated service component.	Resource Offering associated with a Service Component

Internal Action	Description	Applies To
Select Resource Provider from Parent	Selects the resource provider already chosen by a service component's parent service component, as identified by the `Parent Component ID` and `Provider Property Name` properties. The selected resource provider will be available to resource offering actions in the token `RSC_PROVIDER_ID`. The selected resource provider will also be written to a `Provider Property Name` property on the associated service component.	Resource Offering associated with a Service Component

User Operations for Service Components

User operations are actions configured on resource offerings or service, component templates, components in a design that can be invoked by users in the Operations area after a design has been provisioned. For example, a Restart Server user operation can be configured on a server service component or on an associated resource offering, allowing a user to restart a particular provisioned server after the design has been provisioned. User operations can also be invoked on a service instance in the Operations area.

Tasks

> **Note:** If a service design is published, its user operations are read-only.

- **Navigate to** (the **User Operations** tab) - In the left pane of the **All Designs** area, select the tag associated with the design you want to view. Select the service design and, in the **Designer** canvas, select the service component on which you want to complete the task. In the right pane, click the gear icon, select **Edit Component**, and select the **User Operations** tab. From here, you can:

- View a list of user operations for this service component.

- **Add Action** - Click **Add**, and enter the information as described in "User Operation Action Selection Wizard" on page 216.

- **Edit Action** - Select the gear icon on the action you want to edit and select **Edit**. Update the information as described in "User Operation Action Selection Wizard" on page 216.

- **View User Operation** - Click the **User Operations** tab to view all user operations associated with a resource offering.

- **Delete User Operation** - Click the gear icon next to the user operation you want to remove, and select **Delete**.

Action Selection Wizard

Use the action selection wizard to create or edit a user operation.

To add Operations Orchestration flows, see the Import Operations Orchestration Flows section in the *Configuration Guide* for more information.

Tasks

- **Select Action** - Search for flows or actions by name (when searching for a flow, the folders searched in the Operations Orchestration library are determined by a property configured in the `csa.properties` file; see the "Action Selection Wizard" property description section in the Configuration Guide for more information) or select the process engine from which to select a flow or action. Then, locate and select the flow or action.

- **Configure Action** - Configure information about the action.

Item	Description
Process Engine	Read-only. The container of the flow or internal action.
Locator	Read-only. The location of the flow or internal action.
Display Name	The name you provide for the action.
Description	The description you provide for the action.
Execution Properties	Specify the following: ○ Fail on Error - During provisioning or de-provisioning, the Fail On Error setting indicates whether provisioning or de-provisioning should continue after an action has failed. For a user operation, the Fail on Error setting is irrelevant because there are no subsequent actions that would run after the user operation. ○ Error on Timeout - If selected, the user operation will be considered to have failed if the time to execute the action exceeds the Timeout value (see below).
Timeout (seconds)	The amount of time to wait for the action to complete. Set this field to zero (0) if you do not want the action to time out.

- **Configure Parameters** - Configure the input parameters whose values are passed to the flow or internal action prior to invoking the action. Click the gear icon next to the parameter you want to configure and select **Edit Mapping**. From the Edit Parameter Mapping dialog, edit the Name, Display Name, Description, and Mapping Type of the parameter.

Select one of the following Mapping Types:

- Not Mapped - The parameter is not mapped to a value.

- Constant Value - Type a value for the parameter.

- Prompt User - Allows the user to enter a value for the parameter when running the user operation. A default value for the parameter is required.

- Prompt User List - Allows the user to select a value for a parameter by selecting from a list of items when running the user operation.

List Items	
Switch to Static Entry (default)	Enter the display name and value for a list item. After entering a name or value, input boxes for a new list item automatically display.
Switch to Dynamic Entry	- Select a script from the List Items. **HTTP Parameters** - Specify the HTTP parameters to use when executing the script. The required input parameters are dependent on the script that has been selected; contact the script developer if necessary to determine the necessary input parameters. **Refresh Data** - Click the Refresh Data icon ( ) to run the script and display the results.

Associate Resource Offerings with Service Components

For more information about sequenced designs, see "Sequenced Designer" on page 82.

Concepts

You can associate a resource offering with a service component in a service design. For example, a resource offering for a VMware vCenter VM template can be linked to a Server service component. This association or link ensures that the resource offering will be provisioned when the Server service component is deployed.

You can also associate resource offerings with component templates.

Tasks

- **Navigate to** (the Resource Offerings tab) - Select a service design version. Click the **Designer** tab. Select the service component to which you want to add, modify, or delete a resource offering. In the right pane, click the gear icon, select **Edit Component**, and select the **Resource Offerings** tab.

- **Add** - Navigate to the Resource Offerings tab and click **Add**. If no resource offerings are associated, click the **Add** icon in the middle of the window. In the **Select Resource Offering** dialog box, select the **Resource Category** from the dropdown, select **Resource Offering** from the list, and click **Add**.

  A resource offering cannot be associated with a service component if the service component is part of a published design.

- **Delete** - Click the gear icon next to the resource offering you want to delete. Click **Delete** and click **Yes** in the confirmation dialog.

  A resource offering cannot be deleted if the design is in an upgrade path, is upgradable from a previous version of the design, and the resource offering existed in that prior version. Resource offerings that are newly added to the current version of the design can be deleted.

  A resource offering cannot be deleted if the service component is part of a published design.

- **Replace Resource Offering** - Click the gear icon next to the resource offering you want to replace. Select **Replace Resource Offering**, and select the replacement resource offering. The displayed

resource offerings are not bound to the service component and are of the same resource category and provider type as the resource offering being replaced.

- **Navigate to Resource Offering** – Click the gear icon next to the resource offering to which you want to navigate. Select **Navigate to Resource Offering**, which takes you to the resource offering.

- **Group** – Click to select multiple resource offerings, and click **Group** on the right panel to create a parallel processing group. You can also click and drag individual resource offerings to group them. Resource offerings in a parallel processing group are executed at the same time.

- **Ungroup** – Click the gear icon next to a resource offering inside a group and select **Remove from Group**. You can also ungroup a resource offering by dragging it out of the parallel processing group.

  ○ To ungroup multiple resource offerings at one time, click and select multiple resource offerings and click **Ungroup** on the right panel.

- **Set Processing Order** – The processing order of a resource offering specifies the order in which the associated resource offering will be provisioned relative to the other resource offerings added on this service component. The order in which you see the resource offerings are the order in which they will be provisioned. You can click and drag a resource offering or a group of resource offerings and move them around to set processing order.

Provider Selection Panel

For more information about associating resource offerings to service components, see "Associate Resource Offerings with Service Components" on page 105.

Concepts

After adding a resource offering to a service component, you must create one or more provider selection actions for that resource offering. Be sure to do this if you have configured a resource environment in order to control which resource providers should be used when service designs are deployed. Otherwise, the default, random selection of a resource provider that occurs when provider selection actions are not called will ignore the resource environment association. The provider selection actions execute during the Before phase of the Reserving lifecycle stage; the lifecycle stage cannot be changed.

Codar ships with internal actions to help with provider selection. For more information about these provider selection internal actions, see "Provider Selection Internal Actions" on page 109.

Tasks

- **Navigate to** or **View** (the Provider Selection panel) - Select a service design version. Click the **Designer** tab. Select the service component on which you want to add a provider selection action. In the right pane, click the gear icon, select **Edit Component**, and select the **Resource Offerings** tab. In the **Resource Offerings** tab, click the name of the resource offering, and then click **Provider Selection** from the panel on the right.

- **Add** (a provider selection action) - Navigate to the **Provider Selection** panel and click **Add**. If no provider selection actions are configured, click the **Add** icon in the middle of the panel. Enter the information as described in the "Lifecycle Action Selection Wizard" on page 97.

  A provider selection action cannot be created if the service component is part of a published design.

- **Edit** (a provider selection action) - Navigate to the **Provider Selection** panel, click the gear icon next to the action you want to edit, and select **Edit**. Update the information as described in the "Lifecycle Action Selection Wizard" on page 97.

  The properties of an action cannot be edited if the service component is part of a published design.

- **Delete** (a provider selection action) - Navigate to the **Provider Selection** panel, click the gear icon next to the action you want to delete, and select **Delete**.

- **Set Processing Order** – The processing order of a provider selection action is relative to the other provider selection actions on this resource offering. The order in which you see the provider selection actions is the order in which they will be executed. You can click and drag a provider selection action to change its processing order.

Provider Selection Internal Actions

For more information about provider selection actions, see "Provider Selection Panel" on page 107.

Concepts

Codar ships with internal actions to help with provider selection.

- If you have *not* configured resource pools on a provider, then the two provider selection actions most likely to be configured on a resource binding are the **Build Resource Provider List** action and the **Select Resource Provider** action, which should run in that order. To select the provider already selected by the parent service component, use the **Select Resource Provider from Parent** action instead of the two previously mentioned actions.

- If you have configured resource pools on a provider, then the two provider selection actions most likely to be configured on a resource binding are the **Build Resource Provider and Pool List** action and the **Select Resource Provider and Pool** action, which should run in that order. To select the provider and pool already selected by the parent service component, use the **Select Resource Provider and Pool from Parent** action instead of the two previously mentioned actions.

See the following table for descriptions of the out-of-the-box internal actions that ship with Codar:

Internal Action	Description	Applies To
Build Resource Provider and Pool List	Builds a candidate list of resource providers and associated resource pools that meet the following requirements: - Support the resource offering. - Have an **Availability** of **Enabled**. - The provider's resource pool has sufficient resource capacity. To determine this, you must consider all measurable properties as configured in the **Measurable Properties** tab for the resource offerings, as well as the optional **Multiplier Property Name** field. The resource pool must have enough resource capacity to support all the properties, which requires that each necessary resource type (for example: CPU, Memory, and Storage), based on the measurable properties, be configured on the pool either with a **Resource Availability** of **Unlimited** or **Available**. If **Available**, the difference between **Total Available To CSA** and **Current CSA Utilization** must be sufficient to support the measurable property requirements.	Resource Offering associated with a Service Component

Internal Action	Description	Applies To
Build Resource Provider List	Builds a candidate list of resource providers that meet the following requirements: • Support the resource offering. • Have an **Availability** of **Enabled**.	Resource Offering associated with a Service Component
Clone Pattern	Clones a service component that is marked as a **Pattern** into one or more non-pattern service components. The number of service components created is determined by the value of the property specified in **Name of the Property for Service Component Count**.	Service Component
Decrease Resource Utilization	Decreases the utilization of one or more resources in a resource pool by the values of the measurable properties configured on a resource binding. This action should be configured in the **Un-reserve** section of the **Resource Accounting** tab on a resource offering associated with a service component. **Note:** :The Decrease Resource Utilization action cannot be added to a component, component template, or resource offering. This action runs automatically during Un-reserving when Resource Accounting is enabled for a resource offering associated with a service component.	Resource Offering associated with a Service Component
Increase Resource Utilization	Increases the utilization of one or more resources in a resource pool by the values of the measurable properties configured on a resource binding. This action should be configured in the **Reserve** section of the **Resource Accounting** tab on a resource offering associated with a service component.. **Note:** :The Increase Resource Utilization action cannot be added to a component, component template, or resource offering. This action runs automatically during Reserving when Resource Accounting is enabled for a resource offering associated with a service component.	Resource Offering associated with a Service Component
Log Messages	Writes the user-specified Boolean Input, Integer Input, and String Input property values to the `csa.log` file. You can include this as an action on a Service Component, Resource Offering associated with a service component, or Resource Offering for use in troubleshooting.	Resource Offering Resource Offering associated with a Service Component

Internal Action	Description	Applies To
		Service Component
Select Resource Provider	Selects a resource provider from the candidate list that was built by the **Build Resource Provider List** action. The selected resource provider will be available to resource offering actions in the token `RSC_PROVIDER_ID`. The selected provider will, optionally, be written to a property on the associated service component if the `Provider Property Name` input to the action is provided.	Resource Offering associated with a Service Component
Select Resource Provider and Pool	Selects a resource pool and provider from the candidate list that was built by the **Build Resource Provider and Pool List** action. The selected resource provider and pool will be available to resource offering actions in the token `RSC_PROVIDER_ID` and `RSC_POOL_ID`, respectively. The selected pool will, optionally, be written to a property on the associated service component if the `Pool Property Name` input to the action is provided.	Resource Offering associated with a Service Component
Select Resource Provider and Pool from Parent	Selects the resource pool and provider already chosen by a service component's parent service component, as identified by the `Parent Component ID` and `Pool Property Name` properties. The selected resource provider and pool will be available to resource offering actions in the token `RSC_PROVIDER_ID` and `RSC_POOL_ID` respectively. The selected pool will also be written to a `Pool Property Name` property on the associated service component.	Resource Offering associated with a Service Component
Select Resource Provider from Parent	Selects the resource provider already chosen by a service component's parent service component, as identified by the `Parent Component ID` and `Provider Property Name` properties. The selected resource provider will be available to resource offering actions in the token `RSC_PROVIDER_ID`. The selected resource provider will also be written to a `Provider Property Name` property on the associated service component.	Resource Offering associated with a Service Component

Resource Accounting Tab

For more information about associating resource offerings to service components, see "Associate Resource Offerings with Service Components" on page 105.

Concepts

Resource accounting actions track the utilization of resources in a resource pool.

The following out-of-the-box accounting actions are added to the resource offerings when you enable resource accounting:

- **Increase Resource Utilization** - This action is configured to run during the **After** phase of the **Reserving** lifecycle stage.

- **Decrease Resource Utilization** - This action is configured to run during the **After** phase of the **Un-reserving** lifecycle stage.

Tasks

- **Navigate to** or **View** - In the left pane of the **All Designs** area, select the tag associated with the design you want to view, and select the service design. In the **Designer** tab, select the service component whose resource accounting action properties you want to view. In the right pane, click the gear icon and select **Edit Component**. In the **Resource Offerings** tab, select the resource offering. The panel on the right will display information about **Resource Accounting**.

- **Enable** (Resource Accounting) – Navigate to the **Resource Offerings** tab. Select the resource offering on which you want to enable resource accounting. Select the **Enable Resource Accounting** checkbox on the panel on the right. When you enable resource accounting, the two internal actions **Increase Resource Utilization** and **Decrease Resource Utilization** are automatically added to the resource offering.

- **Disable** (Resource Accounting) – Navigate to the **Resource Offerings** tab. Select the resource offering on which you want to disable resource accounting. Deselect the **Enable Resource Accounting** checkbox on the panel on the right. When you disable resource accounting, the two internal actions **Increase Resource Utilization** and **Decrease Resource Utilization** are removed from the resource offering.
  Resource accounting cannot be enabled/disabled if the service component is part of a published design.

Best Practices

- Do not create resource accounting actions on group-level service components. For example, do not create a resource accounting action on a Server Group; instead, create it on a Server service component.

Measurable Properties Tab

For more information about associating resource offerings to service components, see "Associate Resource Offerings with Service Components" on page 105.

Concepts

A measurable property is an integer service component property that has a configured **Resource Type** and **Unit** (see "Service Component Properties" on page 123). When you configure measurable properties on a resource offering, you create references to the corresponding service component measurable properties. To view the value of a measurable property, see the corresponding service component property in the **Properties** tab.

Measurable properties are used by the following out-of-the-box actions to assist in provider and pool selection and in resource accounting:

- Build Resource Provider and Pool List

- Increase Resource Utilization

- Decrease Resource Utilization

If you configure any of these actions on a resource offering in the **Provider Selection** or enabled **Resource Accounting**, you must also configure the measurable properties for that resource offering in the **Measurable Properties** panel.

Tasks

- **Navigate to** or **View** (the Measurable Properties panel) - In the left pane of the **All Designs** area, select the tag associated with the design you want to view. Select the service design and, in the **Designer** tab, select the service component that contains the resource offering whose measurable properties you want to view. In the right pane, click the gear icon and select **Edit Component**. In the **Resource Offerings** tab, click the name of the resource offering whose measurable properties you want to view. Finally, select the **Measurable Properties** panel on the right.

- **Add** - Navigate to the **Measurable Properties** panel and select the measurable properties you want to add. If the service component does not contain any measurable properties, the measurable properties panel will be empty.

A measurable property of a resource offering in a service component cannot be added if it is part of an upgraded or published design.

- **Delete** - Navigate to the Measurable Properties panel and de-select the measurable properties you want to remove.

A measurable property of a resource offering in a service component cannot be deleted if it is part of a published design or a design that is in an upgrade path, is upgradable from a previous version of the design, and the measurable property existed in that prior version.

Change the Service Component Template

You can change the service component template after you add the component to the design. When you change the service component template, the component in the canvas is updated with the following settings from the new template:

- Resource offerings
- Lifecycle actions
- Properties and values
- Display name
- Description
- Image
- Pattern setting
- Consumer-visible setting

For more information about sequenced designs, see "Sequenced Designs" on page 60. For more information about component templates, see "Create a Service Component" on page 86.

To change the service component template

1. From the designer, select the service component to replace.

2. Click the **Select Component Template** icon.

3. Select the service component template to replace the current service component. The replacement component will have the same internal name as the current component, allowing the replacement component to be used in an upgrade path (see "Service Designs in an Upgrade Path" below).

4. Click **Save**.

Service Designs in an Upgrade Path

When replacing a service component in a service design that is part of an upgrade path, the replacement service component template must meet the following criteria in order to replace the current

service component (if one or more of the following criteria are not met, the current service component will not be replaced):

- The *Component Order* in the current service component must be the same in the replacement service component.

- If the *component is a pattern* in the current service component, it must remain as a pattern in the replacement service component. If the *component is not a pattern* in the current service component, it must remain as not a pattern in the replacement service component.

- All *properties* in the current service component must exist in the replacement service component.

- All *property names* in the current service component must be the same in the replacement service component.

- All *resource offerings* in the current service component must exist in the replacement service component.

- All *measurable properties* of all resource offerings in the current service component must exist in the replacement service component.

Duplicate a Service Component

Duplicating a service component creates a copy of the service component in the same design, including its properties, property mappings, resource offerings, and lifecycle actions. The new component also retains the icon image, pattern, execution order, and consumer visible field values of the original component. Any references to properties on the original component are not duplicated to the new clone component.

For information about creating components, see "Create a Service Component" on page 86.

To duplicate a service component

1. Click on the service component you want to duplicate to display the component icon menu.

2. Click the **Duplicate Component** icon .

3. The **Duplicate Component** dialog appears. Enter a **Display Name** for the duplicate component, and click **Save**.

Delete a Service Component

For more information about sequenced designs, see "Sequenced Designs" on page 60.

When you delete a service component, all of its subscriber option target bindings and component property mappings are also deleted.

A service component cannot be deleted if the design is in an upgrade path, is upgradable from a previous version of the design, and the component existed in that prior version. Components that are newly added to the current version of the design can be deleted.

To delete a service component

1. Select the service component you want to delete.

2. Click the icon 🗑.

3. Click **Yes** to confirm the deletion.

Create a Property Mapping

For more information about sequenced designs, see "Sequenced Designer" on page 82.

Concepts

You can create a property mapping between the properties of two components in a design when the property of one component should obtain its value from the property of another component.

Using property mapping, you can share a single property value between multiple components in a design without the need to duplicate the value of the property across those components. For example, if your design includes three components that represent servers, and you need each server to be provisioned with the same number of CPUs, you can define the number of CPUs on one component and then create a property mapping between the CPU properties of the other components and the component that stores that common value.

A property mapping can only be created between properties of the same type.

A property mapping cannot be created between two properties on the same component, or where the mapping creates a loop. For example, a property mapping where property A is mapped to obtain its value from property B, which is mapped to obtain its value from property A is not valid.

Tasks

Property mappings are managed in the Designer tab of a service design version. You can perform the following tasks in the **Designer** tab:

- **Create** - Select the service component to which you want to add a property mapping. In the component properties pane, click the gear icon next to the property you want to map and then select **Map Property**. In the Select Mapping dialog, select a different component that has compatible properties to create the mapping. Select a compatible property of the selected component.
The flow of the property mapping is displayed at the bottom of the Select Mapping dialog.

- **Delete**- Click the gear icon next to the mapped property you want to delete and then select **Delete Mapping**.

- **View** - Select the service component with the property mapping you want to view. Mapped properties are displayed with the chain icon in the properties pane on the right.

- **Edit** - In the properties pane, click the gear icon next to the property mapping you want to update and then select **Update Mapping**. The component with the property mapping is highlighted in the Select Mapping dialog. The properties of the mapped component are displayed in the right pane. The mapped property is displayed at the bottom of the Select Mapping dialog. Click a property on the right pane to create a mapping to a different property of the mapped component. You can also select a different component and its property to create a mapping in the Select Mapping dialog, or click the trashcan icon to delete a property mapping.

Delete a Relationship between Service Components

A relationship between two service components cannot be deleted if the design is in an upgrade path, is upgradable from a previous version of the design, and the relationship between the components existed in that prior version.

A relationship between two components that is newly added to the current version of the design can be deleted.

To delete a relationship between service components

1. Click the relationship between two service components that you want to delete.

2. Click the trashcan icon .

3. Click **Yes** to confirm the deletion.

Service Component Properties

For more information about service components, see "Service Components" on page 85. For more information about component type properties, see "Create Component Type Properties" on page 187

Service component properties are managed using the **Properties** panel and the **Edit Component** menu in the **Designer** tab.

For more advanced configuration of the properties on a component, including the ability to add or delete properties, use the **Edit Component** menu. For more information, see "Edit a Service Component" on page 88.

Concepts

Properties for service components are user-defined properties that you may want to create in the following situations:

- When an action that runs on a service component or its associated resource offering expects a property value with a given name/value to exist on the service component. For example, an action may need to retrieve an IP Address that is stored on a Server service component.

- When you want to create a property mapping between components in a design, where the property of a component obtains its value from the property of another component in a design. For example, a Software Application Tier service component may need to know the service component ID of a Server Group. You can create a property on the Server Group service component called `SVC_COMPONENT_ID`, which has a token value of `[TOKEN:SVC_COMPONENT_ID]`. You would also create a property on the Software Application Tier service component whose value contains a property mapping to the Server Group property (`SVC_COMPONENT_ID`). The property value is then pulled by the Software Application Tier from that `SVC_COMPONENT_ID` property on the Server Group. This use of service component properties to pull a value from another service component property is called a *property mapping*. Another type of relationship between properties is called a *target binding*.

- When you want to expose a property value in the Marketplace Portal. The property value can be explicitly defined on the service component, or may be set as the result of an action execution (for example, a HOSTNAME property might be set on a Server service component as part of deploying the Server).

  > **Note:** Marketplace Portal is available only when the CSA license is applied.

Tasks

- **View** - Select the service component whose properties you want to view. The **Properties** pane opens on the right and displays the properties of the selected component. A property that is set as a measurable property is displayed with its unit in parenthesis. The component type and the palette to which the component belongs are displayed at the top of the **Properties** pane. Click the **References** link under a property to view component and subscriber option property mappings.

- **Edit** - Select the service component whose properties you want to edit. The **Properties** pane opens on the right and displays the properties of the selected component. You can edit the values of any property and click **Save** to persist the changes.

- **Map properties** - Create a mapping between the properties of components in a design. For more information, see "Create a Property Mapping" on page 120

- **Add, edit, or delete list items** - Add, edit, or delete list items for list properties. Use the **Edit Component** menu to add list properties.

- **Configure advanced settings** - Add, edit, delete, or modify properties of a service component, configure lifecycle actions, and configure resource offerings. For more information, see "Edit a Service Component" on page 88.

Service Design Deployments

You must be in the administrator or designer role to view service design instances and deployments. The administrator can see all instances and deployments for a design; the designer can only see the instances and deployments he or she created. For more information about roles, see Roles in .

1. Select a version of a service design (see "View Service Design and View Service Design Version" on page 64).

2. Click the **Deployments** tab to see all service instances and deployments for this version of the design.
   Enter text in the search box to filter deployments by name or description.

3. Click on a deployment name to view additional information about the deployment. See the Operations area for additional information.

Create and Deploy a New Instance

You can create and deploy a new service instance from a service design. For more information about sequenced designs, see "Sequenced Designs" on page 60.

1. In the Deployments area, click **Deploy a New Instance**. The dialog appears. Enter the following general information:

   a. **Deployment Name** - Enter a deployment name. The service instance created from this action will have the same name as the deployment.

   b. **Description** (optional) - Enter a description of the deployment.

   c. **Resource Environment Filter** - From the list, select one or more resource environments to use for provisioning. For more information, see Resource Offerings (Sequenced Designs).

2. Enter properties for the new instance:

   ○ The design associated with the instance appears on the canvas. Select a service component from the design, and modify the properties as needed. For more information, see "Service Components" on page 85.

   ○ You can change the properties of multiple components in the design. The message icon appears next to modified components in the design. Click on the icon for details about what changes were made.

3. Click **Deploy**.

Run a User Operation from a Component in a Deployment

You can run a user operation on a component in an active deployment.

1. Navigate to a deployed instance in a service design.

2. Click on the Topology tab, and select a component from the canvas or the list on the right panel.

3. From the bottom of the right panel, click Actions, and use the list to select a user operation to run.

4. A list of the properties for the user operation appears. Modify the properties or accept the defaults, and click Execute Action. You can execute another user operation when the first one completes.

See ""User Operations for Service Components" on page 102" for information on how to create a user operation.

Cancel a Deployment

1. Click the gear icon to the right of the deployment you want to cancel, and select **Cancel Deployment**.

2. Click **Yes** to confirm the cancellation.

A canceled deployment is moved to **Offline** state.

Delete a Deployment

When you delete a deployment, it is removed from the list on the **Deployments** tab, and all of its references are marked internally as being in **Retired** state. Once in a **Retired** state, a deployment can be deleted permanently from the database using the purge tool. For more information about purge tool usage, see the Configuration Guide.

To delete a deployment

1. Click the gear icon to the right of the deployment in **Offline** state that you want to delete, and select **Delete Deployment**.

2. Click **Yes** to confirm the deletion.

Manage Tags

To manage tags

Tags are a label you can use to provide a structure for organizing and grouping related items.

1. From the All Designs area, click the gear icon and select **Manage Design Tags**.

2. In the **Manage Tags** dialog, you can:

- "Create a Tag" on the next page

- "Edit a Tag" on page 130

- "Delete a Tag" on page 131

Create a Tag

For more information about tags, see "Manage Tags" on the previous page.

To create a tag

1. In the Manage Tags dialog, click the **Add** icon.

2. In the Create Tag dialog, enter the following information:
    a. **Display Name** - The name you provide for this tag.

    b. **Description** - The description you provide for this tag.

    c. **Image** - An image that displays for the tag. Click **Change Image**. Choose the image you want, and click **Select**. Click **Upload** to add your own image. Supported file extensions include `.jpg`, `.jpeg`, `.gif`, and `.png`. The recommended image size is 256 by 256 pixels, and the image will be scaled to the appropriate size. The images are stored in the `%CSA_HOME%\jboss-as\standalone\deployments\csa.war\images\library` folder of the Codar server.

    d. **Color** - A color that displays for the tag.

3. Click **Create**.

Edit a Tag

For more information about tags, see "Manage Tags" on page 128.

To edit a tag

1. In the Manage Tags dialog, click the **Edit** icon.

2. Make your changes and then click **Save**. For information about certain fields, see "Create a Tag" on the previous page.

Delete a Tag

For more information about tags, see "Manage Tags" on page 128.

To delete a tag

1. In the Manage Tags dialog, select the tag you want to delete.

2. Click the **Delete** icon.

3. Click **Yes** in the confirmation dialog.

Resource Offerings Overview Tab

The **Overview** tab displays the following information about the offering:

Item	Description
Display Name	Resource offering name.
Description	Resource offering description.
Provider type	Provider associated with the resource offering.
Resource category	Category associated with the resource offering.
Image	Image associated with the resource offering.

The following options are also available from the **Overview** details page, in the gear icon list:

Edit a resource offering

To edit the resource offering, in the resource offering's details page, click the gear icon and choose **Edit**. Edit the display name or description and click **Save**.

Delete a resource offering

To delete the resource offering, in the resource offering's details page, click the gear icon and choose **Delete**.

Warning: An offering cannot be deleted if it is used in a service design. When an offering is deleted, its associations to providers are automatically removed.

Copy a resource offering (Save As)

To copy a resource offering, in the resource offering's details page, click the gear icon and choose **Save As**. Enter a display name and description and click **Save**.

Export a resource offering

To export a resource offering, in the resource offering's details page, click the gear icon and click **Export**.

See Import and Export a Resource Offering for more information on the Export option.

Resource Offering - Tabs

A resource-offering landing page has the following tabs:

- Overview tab - Displays the resource offering's name and details, and allows you to edit, copy, export, or delete the resource offering.

- Provider tab - Displays information about current providers associated with the resource offering. The tab also allows you to create a provider association, and to select the provider.

- Lifecycle tab - Displays the resource offering lifecycles and allows you to chose or select a stage, and edit, delete, save as, or export a resource offering at a particular stage.

- User Operations tab - Displays user operations associated with a resource offering, and allows you to add, edit, delete, copy, or export resource offerings associated with a user operation.

- Component Templates tab - Allows you to view component templates associated with a resource offering, and edit, copy, delete, and export the resource offering associated with the template.

- Service Designs tab - Allows you to view service designs associated with the resource offering, edit associated service design information, and create or delete service-design associations

Resource Offerings - Providers Tab

This topic is divided into **Concepts** and **Tasks**.

Concepts

Providers are management platforms that offer centralized control over the infrastructure and resources that a cloud computing environment uses. One provider can deploy virtual machines, while another provider can monitor applications. A provider corresponds to the specific instance of an application, and integrates with the application help instantiate service designs.

A **provider type** allows you to classify providers for improved filtering and identification, and includes some predefined, out-of-the-box provider types.

Each instance of a resource offering can have a single provider type. In addition, resource offerings can be associated only with providers that share the same provider type.

After adding a resource offering to a service component, you must create one or more provider selection actions for that resource offering. The provider selection actions execute during the Before phase of the Reserving lifecycle stage; the lifecycle stage cannot be changed.

Codar ships with internal actions to help with provider selection. For more information about these provider selection internal actions, see "Provider Selection Internal Actions" on page 109.

Tasks

The resource offering **Provider tab** displays information about current providers associated with the resource offering, and is accessible from the resource offering's details page.

You can perform the following actions in the Provider tab gear:

- **View details** - Click a provider **name** to view provider details.
- **Delete** - Chose **delete** to delete the resource offering.
- **Select** - Click **Select Providers** to display a list of available providers that you can associate to a resource offering.

Resource Offerings Properties Tab

In the Properties tab, in the resource offering's details page gear, you can perform the following actions:

- **Create a property** - Click **Create**.

In addition to a name and display name, enter the following information in the **Create Property** wizard:

Item	Description
Property Type	Select one of the following: • **Boolean** - A property whose value is true or false. • **List** - A property whose value is a list of String values. • **Integer** - A property whose value is a positive or negative whole number or zero. • **String** - A property whose value is a sequence of characters. You cannot edit this value after the property has been created.
Property Details	*For Boolean properties:* • **Name** - A unique name for the property. • **Display Name** - The display name for the property. • **Description** - A description of the property. • **Property Value** - Select **true** or **false**. *For List properties:* • **Name** - A unique name for the property. • **Display Name** - The display name for the property. • **Description** - A description of the property. • **Value Entry Method**: - Select **Manual Entry** to provide **Specified Values** for this field, as described below. • **Specified Values** - Click the **Add Value** icon to add a new value, or click the **Remove Selected Value(s)** icon to remove a selected value. *For Integer properties*: • **Name** - A unique name for the property. • **Display Name** - The display name for the property.

Item	Description
	• **Description** - A description of the property. • **Property Value** - Select or type a positive or negative whole number or zero. If you enter a decimal number, the value will be truncated to the nearest integer. The maximum allowed integer value is 2147483647 and the minimum is -2147483648; if you enter a value outside these bounds the value will be automatically converted to the closest maximum or minimum value. *For String properties:* • **Name** - A unique name for the property. • **Display Name** - The display name for the property. • **Description** - A description of the property. • **Property Value** - Type a string of characters. • **Confidential Data** - Select this box to mask the values so that they cannot be read in the user interface; no encryption of the value is performed.

- **Edit a property** - Chose **Edit**. You can edit a property's display name, description, and property value. Click **Save**.
- **Delete a property** - Chose **Delete**.

Lifecycle Actions for Resource Offerings

This topic has two sections: **Concepts** and **Tasks**.

Concepts

What is a resource offering lifecycle?

A resource offering lifecycle is a collection of actions defined for a resource offering. The Lifecycle area allows you to specify the actions that are needed to provision, update, and deprovision resource offerings. Use the action selection wizard to add a lifecycle action to a resource offering from one of the available process engines.

What is a lifecycle action?

A lifecycle action is a function that is either run automatically at a specified lifecycle stage or phase, or that is exposed to the user. Most lifecycle actions correspond to OO flows, which contain the logic for executing the function. Codar also includes actions that run internal to Codar. Many Codar internal actions relate to provider and pool selection and resource accounting. Actions include input parameters that provide configuration information to the function.

What are lifecycle stages?

A lifecycle stage represents a step within the Codar service lifecycle. There are three categories of lifecycle stages: Provisioning, Operational, and De-provisioning.

The Provisioning lifecycle stages are:

- **Initializing** - Initializing is the first stage that is processed during provisioning. Use the Initializing stage for actions that perform any type of initialization that is required before proceeding to the Reserving stage, such as actions that perform input validation or create change request records.

- **Reserving** - Reserving is the second stage that is processed during provisioning, after the Initializing stage. When a resource offering is associated to a service component, any configured provider selection and pool selection actions run during the Reserving stage. If resource accounting is enabled, actions to reserve resources from a pool run during the Reserving stage. In general, any resources needed during provisioning, such as storage or networking, can be reserved during the Reserving stage.

- **Deploying** - Deploying is the final stage that is processed during provisioning, after the Initializing and Reserving stages, and is the most common stage in which provisioning actions are defined. Define actions in the Initializing and Reserving stages to prepare the environment for provisioning.

Define actions in the Deploying stage to perform actual provisioning. For example, you can define actions that deploy a VM or an application in the Deploying stage.

The De-provisioning lifecycle stages are:

- **Un-deploying** - Un-deploying is the first stage that is processed during de-provisioning and is the most common stage in which de-provisioning actions are defined. Use the Un-deploying stage to undo or to finalize the actions that took place during the Deploying stage. For example, if actions in the Deploying stage deploy a VM, define actions in the Un-deploying stage to un-deploy the VM.

- **Un-reserving** - Un-reserving is the second stage that is processed during de-provisioning, after the Un-deploying stage. Use the Un-reserving stage to undo or to finalize the actions that took place during the Reserving stage. For example, if resource accounting is enabled, actions to return resources to the pool run during the Un-reserving stage.

- **Un-initializing** - Un-initializing is the final stage that is processed during de-provisioning, after the Un-deploying and Un-reserving stages. Use the Un-initializing stage to undo or to finalize the actions that took place during the Initializing stage. For example, if actions in the Initializing stage create a change request, define actions in the Un-initializing stage to finalize that change request.

What are lifecycle phases?

A lifecycle phase is a further refinement of a lifecycle stage. When defining a lifecycle action at a lifecycle stage, you can also specify the phase for the action. The phases are:

- **Before** - Actions configured in the Before phase on a service component run before all actions on child service components in a design and before all actions on any associated resource offerings. Actions configured in the Before phase on a resource offering run after all of the Before actions on the associated service component have completed.

- **During** - Actions configured in the During phase on a service component run after all of the actions have completed on child service components in a design but before any of the During actions run on any associated resource offerings. Actions configured in the During phase on a resource offering run after all of the During actions on the associated service component have completed.

- **After** - Actions configured in the After phase on a service component run after all actions have completed on child service components and after all of the During actions have completed on the service component and any associated resource offerings. Actions configured in the After phase on a resource offering run after all of the After actions on the associated service component have completed.

- **On Failure** - Actions configured in the On Failure phase on a service component or resource offering run after an action failure occurs in any of the Before, During, or After phases for that service component or resource offering.

Tasks

- **Navigate to** (the Lifecycle tab) - In the left pane of the **Resource Offerings** area, select the category or provider type associated with the resource offering you want to view. Select the resource offering and then select the **Lifecycle** tab.

- **Add Action** - Click the **Add Action** or **Add More Actions** link in the stage and phase to which you want to add an action. Enter the information as described in "Lifecycle Action Selection Wizard" on page 153.

- **Edit Action** - Select the gear icon on the action you want to edit and select **Edit**. Update the information as described in "Lifecycle Action Selection Wizard" on page 153.

- **Delete Action**- Click the gear icon next to the action you want to delete and select **Delete**. You can also delete an action that is in a group.

- **Move Action**- Select the gear icon on the action or group of actions you want to move to another stage or phase and select **Move**. In the Move dialog, select the Target Stage and Target Phase from the pull-down menus and then click **Move**. You can also click and drag to move an action or group of actions. To move an action or group of actions using the keyboard, press the tab key to provide focus on the action or group and then press Ctrl and the up or down keys to move the action or group within a phase.

- **Group Actions** - Click to select multiple actions in a lifecycle phase and then click **Group Actions** to create a parallel execution group. Actions in a parallel execution group are executed at the same time.

- **Ungroup Actions** - Click the gear icon next to an action in a group and then select **Remove from Group**.

- **View Lifecycle Stages** - The Deploying Provisioning stage, the Modifying Operational stage the Un-deploying De-provisioning stage, and any stage that has an action, are displayed by default. Select a filter from the All Selected Stages pull-down menu to display only Provisioning Stages, Operational Stages, or De-provisioning stages. Click **Select Stages** to view additional stages.

- **View Lifecycle Phases** - The During and On Failure phases are displayed by default. Select **Show All Phases** to display the Before and After phases.

User Operations for Resource Offerings

This topic has two sections: **Concepts** and **Tasks**.

Concepts

User operations are actions configured on resource offerings or service, component templates, components in a design that can be invoked by users in the Operations area after a design has been provisioned. For example, a Restart Server user operation can be configured on a server service component or on an associated resource offering, allowing a user to restart a particular provisioned server after the design has been provisioned. User operations can also be invoked on a service instance in the Operations area.

Tasks

- **Navigate to** (the **User Operations** tab) - In the left pane of the **Resource Offerings** area, select the category or provider type associated with the resource offering you want to view. Select the resource offering and then select the **User Operations** tab.

- **Add Action** - Click **Add**, and enter the information as described in "User Operation Action Selection Wizard" on page 216.

- **Edit Action** - Select the gear icon on the action you want to edit and select **Edit**. Update the information as described in "User Operation Action Selection Wizard" on page 216.

- **View User Operation** - Click the **User Operations** tab to view all user operations associated with a resource offering.

- **Delete User Operation** - Click the gear icon next to the user operation you want to remove, and select **Delete**.

Resource Offerings - Component Template Tab

This topic contains two sections: **Concepts** and **Tasks**.

Concepts

Components are elements of service design, sequenced or topological. A component template is a specialized version of a component type and is used to simplify service-design creation. Component templates include customized settings for properties, lifecycle actions, and resource offerings normally created in a service design.

Sequenced components are not associated with providers or provider types. From the Components tab, you can view the topological components associated with a specific provider instance and manage the topological components.

Note: Provider components are applicable only to topology components and are not applicable to sequenced components.

Tasks

The **Component Templates tab**, in the resource-offering's details page, allows you to perform several tasks:

- **View list** - View a list of component templates associated with a resource offering.

- **Select a component template** - Click a component-template name to select it and display its details page.

See View Component Templates to add, edit, copy, or delete a component template.

Resource Offerings - Service Designs Tab

This topic consists of the following sections: **Concepts** and **Tasks**.

Concepts

To provide on-demand, automated service delivery, you create, configure, and modify service designs. A service design is a template (or blueprint) for an orderable service. Service designs include a hierarchy of service components, resource bindings, lifecycle actions, and custom properties, as defined by the Service Designer. Service components and their relationships in a service design define the framework for creating the service.

Service designs also provide a structure for options that consumers can select when ordering a service. You can also leverage service designs shipped with Codar as well as exporting and importing designs between Codar systems.

Tasks

The **Service Designs tab**, in the resource-offering's details page, allows you to perform several tasks:

- **View list** - View a list of service designs that use the resource offering.

- **Select a service design** - Click a service-design name to select it and to display its details page in **My Applications > Designs > Sequenced > Designer > Sequenced Designs**.

See "View Service Design and View Service Design Version" on page 64 to add, edit, copy, or delete a service design.

View Resource Offerings

For more information about resource offerings, see "Resource Offerings".

To view resource offerings

1. Select **Provider Type** or **Category** from the list on the left to see all the resource offerings in the system.

2. **Provider Type** is selected by default. The left panel shows all provider types in the system. All resource offerings for the provider types appear in the right panel.

3. When **Category** is selected, the left panel shows all categories in the system. All resource offerings for the categories appear in the right panel.

4. For descriptions of the specific properties, see "Create a Resource Offering" on the next page.

Create a Resource Offering

To create a resource offering, in the resource offerings summary page, click the gear icon and choose **Create Resource Offering**. Enter the following information:

Item	Description
Display Name	The name you provide for the offering.
Description	The description you provide for the offering.
Provider Type	The provider type for this offering. This field cannot be changed after an offering is created.
Resource Category	The resource category for this offering. For more information, see "Resource Offering Categories" on page 169. This field cannot be changed after an offering is created.
Image	The image of the provider type selected. This field cannot be changed.

Click **Create**.

Import and Export a Resource Offering

For more information about resource offerings, see "Resource Offerings".

Concepts

In this topic, an artifact refers to a resource offering.

You can import and export many of the artifacts that provide the basis for cloud automation. The export operation provides the ability to preserve the selected artifacts so they can be used to replicate the services on another system or to restore the artifacts. These exported archive files are preserved in an industry-standard zip archive file format.

Installing or replacing artifacts on the target system is supported by import and update operations. The import operation only adds artifacts, whereas the update operation replaces matching artifacts. See the *Tasks* section below for more information.

Process Information

- **Resource categories and provider types** - Resource categories (such as Compute) and provider types (such as VMware vCenter) are resolved first by **name** and second by **display name** during import of a resource offering.

  Out-of-the box resource categories and provider types have identical **name** values on all installations of Codar, and automatically resolve correctly during import. User-created resource categories and provider types do not have a **name** match on different installations of Codar, and instead are resolved by **display name**.

  For example, if a user created resource category with a display name of Auditing is used for a resource offering, when that resource offering is imported on another Codar installation, an attempt will be made to match a resource category with a **display name** of Auditing. This match is successful only if the user has already created this corresponding resource category on the system in which the import occurs.

  If a resource category or provider type cannot be resolved by either **name** or **display name**, a new resource category or provider type is automatically created during import of the resource offering. There is no need to add user-created resource categories and provider types ahead of time on the import system; however, if you have done so, ensure the **display name** values used match those on the export system.

- **Update operation** - The update operation is destructive to existing data. You should understand the differences between the import operation, update operation, and update options to ensure you choose the operations and options that match your expectations. See the *Tasks* section below for more information.

- **Flows** - During an import, update, or preview operation, if required dependencies do not exist on the Operations Orchestration system, an error message identifies these missing dependencies (dependencies such as flows). The content pack that contain these flows must have been deployed to the Operations Orchestration system prior to importing these artifacts. The flows must also have identical signatures and identical paths as the flows on the system from which the artifact was exported.

  During import, flow signature-related information is verified in or added to the Codar database (flow signatures are used during the creation of an artifact and when adding a resource synchronization action or an external approval type). This information is resolved by **name** which corresponds to the full path to the Operations Orchestration flow (for example, `/Library/CSA Content Pack/CSA3.2/Providers/Infrastructure/vCenter/vCenter Clone Server/Actions/vCenter Simple Compute - Deploy`). For information about how to deploy Operations Orchestration content packs, see the *Central User Guide*.

- **Archive content** - The archive (`.zip`) file can only reference files or content contained within the `.zip` file itself, or that are already contained in the `csa.war` file.

- **Image files** - All images in the archive files must end in one of the following suffix values (for information on adding additional suffix values, see the *Configuration Guide*):

  `jpg|jpeg|jpe|jfif|svg|tif|tiff|ras|cmx|ico|pnm|pbm|pgm|ppm|rgb|xbm|xpm|xwd|png`

- **Back up your data** - Create a backup of your system or data. Before proceeding, be sure to create a backup of any artifacts you may be affecting by using the export operation to save an archive zip file.

Tasks

- **Import** option - Adds only new artifacts to the target system (the archived artifacts do no exist on the target system).

  Note: The import operation identifies resource offerings by functional equivalence to determine if an identical resource offering exists on the target system. An archived resource offering is considered to be functionally equivalent to the resource offering on the target system if the archived resource offering has the same resource category, provider type, properties, and

> actions as a resource offering on the target system. If the import operation identifies a functionally equivalent resource offering on the target system, the archived resource offering is not imported. See How Codar identifies matching artifacts for more information.

- **Update** option - Imports new artifacts and overwrites existing artifacts. New artifacts are created if they do not exist on the target system. Identical artifacts that exist on the target system are overwritten with changes from the archive.

  Preserve Originals - Select this option to create a copy of the original artifact if the artifact exists on the target system. On import, a copy of the original artifact is created and then the original artifact is overwritten by the imported artifact. The version of the copy is appended with "Superseded on" and the date. The version of the artifacts being imported remain intact. If the artifact does not exist on the target system, a copy is not created.

  > **Note:** The update operation identifies resource offerings by name instead of functional equivalence to determine if the resource offering exists on the target system.

- **Preview** - Generate a report of prospective results for the import process, including information about the artifacts and their status.

- **View Detailed Report** - From the Import Summary dialog, select this link to see a summary and details of the import process, including information about the artifacts and their status.

- **Export** - Create a content archive (.zip) file. The content archive contains XML documents for the artifacts you are exporting, as well as icons for customizing the artifacts, and the Manifest XML document, which contains meta-information about the archive files.

  The resource offerings are packaged in an archive file whose name is:

  ```
  RESOURCE_OFFERING_<resource_offering_display_name>_<resource_offering_id>.zip
  ```

How Codar identifies matching artifacts

When you import a resource offering, Codar determines if a functionally equivalent resource offering already exists on the system. If a functionally equivalent resource offering exists, then the import may be skipped depending on the options selected. Functional equivalence is determined by comparing the imported resource offering with other resource offerings that exist on the system, as follows:

- At a high level, two resource offerings are functionally equivalent if they share the same Provider Type and Resource Category, the same set of user defined custom properties, and the same set of lifecycle actions.

- Properties are considered equivalent if they share the same Type, Name, and Value (or Values for List properties).

- Lifecycle actions are considered equivalent if they share the same internal action or flow, lifecycle stage and phase, execution order, and action input properties and values.

- There are some additional attributes on properties and lifecycle actions that must be identical for equivalence.

For more specifics on the precise requirements for resource offering equivalence as it relates to entries in the .zip file produced during resource offering export, see the following table:

Element	Necessary for Equivalence
property	- name - valueType - values - confidential – only for String property types
action	- processDefinition name - lifecycleState name - lifecycleSubstate name - lifecycleExecOrder - errorOnTimeout - failOnError - timeout - all properties must be identical, including the consumerVisible and consumerReadOnly elements for each property - consumerVisible
resourceCategory	isCriticalSystemObject determines if this is an out -of-box resourceCategory. If true, name determines equivalence, otherwise displayName determines equivalence.
providerType	isCriticalSystemObject determines if this is an out -of-box providerType. If true, name determines equivalence, otherwise displayName determines equivalence.

For more information about importing and exporting artifacts, see the *Cloud Service Automation Content Archive Tool* document.

Select or Remove Providers with a Resource Offering

For more information about resource offerings, see Resource Offerings.

For a resource offering to be successfully provisioned at design deployment time, at least one resource provider must be associated with the resource offering.

When you associate a resource offering with a provider, the following must be true for successful provisioning:

- The associated providers support the resource offering and can deploy it when a service instance is provisioned.

- If a resource offering is associated with multiple providers, the resource offering must be exactly the same on each of the associated providers. For example, the VMware vCenter template must be exactly the same on all the associated providers.

Select providers for a resource offering

You must select at least one provider for a resource offering if you wan to use the resource offering in service designs.

To select a provider, in the resource offering's details page, click the **Providers** tab, then click **Select Providers**. Click the provider name, then click **Select**.

Delete providers from a resource offering

To delete providers from a resource offering, in the resource offering's details page, click the **Providers** tab. Click the provider's gear icon and click **Remove**.

> **Note:** This action deletes the provider from the resource offering, but does not delete the provider itself from Codar.

Lifecycle Actions for Resource Offerings

For more information about resource offerings, see Resource Offerings.

Concepts

What is a resource offering lifecycle?

A resource offering lifecycle is a collection of actions defined for a resource offering. The Lifecycle area allows you to specify the actions that are needed to provision, update, and deprovision resource offerings. Use the action selection wizard to add a lifecycle action to a resource offering from one of the available process engines.

What is a lifecycle action?

A lifecycle action is a function that is either run automatically at a specified lifecycle stage or phase, or that is exposed to the user. Most lifecycle actions correspond to OO flows, which contain the logic for executing the function. Codar also includes actions that run internal to Codar. Many Codar internal actions relate to provider and pool selection and resource accounting. Actions include input parameters that provide configuration information to the function.

What are lifecycle stages?

A lifecycle stage represents a step within the Codar service lifecycle. There are three categories of lifecycle stages: Provisioning, Operational, and De-provisioning.

The Provisioning lifecycle stages are:

- **Initializing** - Initializing is the first stage that is processed during provisioning. Use the Initializing stage for actions that perform any type of initialization that is required before proceeding to the Reserving stage, such as actions that perform input validation or create change request records.

- **Reserving** - Reserving is the second stage that is processed during provisioning, after the Initializing stage. When a resource offering is associated to a service component, any configured provider selection and pool selection actions run during the Reserving stage. If resource accounting is enabled, actions to reserve resources from a pool run during the Reserving stage. In general, any resources needed during provisioning, such as storage or networking, can be reserved during the Reserving stage.

- **Deploying** - Deploying is the final stage that is processed during provisioning, after the Initializing and Reserving stages, and is the most common stage in which provisioning actions are defined. Define actions in the Initializing and Reserving stages to prepare the environment for provisioning.

Define actions in the Deploying stage to perform actual provisioning. For example, you can define actions that deploy a VM or an application in the Deploying stage.

The De-provisioning lifecycle stages are:

- **Un-deploying** - Un-deploying is the first stage that is processed during de-provisioning and is the most common stage in which de-provisioning actions are defined. Use the Un-deploying stage to undo or to finalize the actions that took place during the Deploying stage. For example, if actions in the Deploying stage deploy a VM, define actions in the Un-deploying stage to un-deploy the VM.

- **Un-reserving** - Un-reserving is the second stage that is processed during de-provisioning, after the Un-deploying stage. Use the Un-reserving stage to undo or to finalize the actions that took place during the Reserving stage. For example, if resource accounting is enabled, actions to return resources to the pool run during the Un-reserving stage.

- **Un-initializing** - Un-initializing is the final stage that is processed during de-provisioning, after the Un-deploying and Un-reserving stages. Use the Un-initializing stage to undo or to finalize the actions that took place during the Initializing stage. For example, if actions in the Initializing stage create a change request, define actions in the Un-initializing stage to finalize that change request.

What are lifecycle phases?

A lifecycle phase is a further refinement of a lifecycle stage. When defining a lifecycle action at a lifecycle stage, you can also specify the phase for the action. The phases are:

- **Before** - Actions configured in the Before phase on a service component run before all actions on child service components in a design and before all actions on any associated resource offerings. Actions configured in the Before phase on a resource offering run after all of the Before actions on the associated service component have completed.

- **During** - Actions configured in the During phase on a service component run after all of the actions have completed on child service components in a design but before any of the During actions run on any associated resource offerings. Actions configured in the During phase on a resource offering run after all of the During actions on the associated service component have completed.

- **After** - Actions configured in the After phase on a service component run after all actions have completed on child service components and after all of the During actions have completed on the service component and any associated resource offerings. Actions configured in the After phase on a resource offering run after all of the After actions on the associated service component have completed.

- **On Failure** - Actions configured in the On Failure phase on a service component or resource offering run after an action failure occurs in any of the Before, During, or After phases for that service component or resource offering.

Tasks

- **Navigate to** (the Lifecycle tab) - In the left pane of the **Resource Offerings** area, select the category or provider type associated with the resource offering you want to view. Select the resource offering and then select the **Lifecycle** tab.

- **Add Action** - Click the **Add Action** or **Add More Actions** link in the stage and phase to which you want to add an action. Enter the information as described in "Lifecycle Action Selection Wizard" on the next page.

- **Edit Action** - Select the gear icon on the action you want to edit and select **Edit**. Update the information as described in "Lifecycle Action Selection Wizard" on the next page.

- **Delete Action**- Click the gear icon next to the action you want to delete and select **Delete**. You can also delete an action that is in a group.

- **Move Action**- Select the gear icon on the action or group of actions you want to move to another stage or phase and select **Move**. In the Move dialog, select the Target Stage and Target Phase from the pull-down menus and then click **Move**. You can also click and drag to move an action or group of actions. To move an action or group of actions using the keyboard, press the tab key to provide focus on the action or group and then press Ctrl and the up or down keys to move the action or group within a phase.

- **Group Actions** - Click to select multiple actions in a lifecycle phase and then click **Group Actions** to create a parallel execution group. Actions in a parallel execution group are executed at the same time.

- **Ungroup Actions** - Click the gear icon next to an action in a group and then select **Remove from Group**.

- **View Lifecycle Stages** - The Deploying Provisioning stage, the Modifying Operational stage the Un-deploying De-provisioning stage, and any stage that has an action, are displayed by default. Select a filter from the All Selected Stages pull-down menu to display only Provisioning Stages, Operational Stages, or De-provisioning stages. Click **Select Stages** to view additional stages.

- **View Lifecycle Phases** - The During and On Failure phases are displayed by default. Select **Show All Phases** to display the Before and After phases.

Lifecycle Action Selection Wizard

See also Lifecycle Action Selection Wizard (for service components) or Lifecycle Action Selection Wizard (for service component templates).

Concepts

Use the action selection wizard to create or edit an action for the selected lifecycle phase.

To add Operations Orchestration flows, see the Import Operations Orchestration Flows section in the *Configuration Guide* for more information.

Tasks

- **Select Action** - Search for flows or actions by name (when searching for a flow, the folders searched in the Operations Orchestration library are determined by a property configured in the `csa.properties` file; see the "Action Selection Wizard" property description section in the Configuration Guide for more information) or select the process engine from which to select a flow or action. Then, locate and select the flow or action. See "Lifecycle Action Internal Actions" on page 155 for a list of internal actions. This task is only available when you create an action.

- **Configure Action** - Configure information about the action.

Item	Description
Process Engine	Read-only. The container of the flow or internal action.
Locator	Read-only. The location of the flow or internal action.
Display Name	The name you provide for the action.
Description	The description you provide for the action.
Execution Properties	Specify the following: ○ Fail on Error - If selected, provisioning or de-provisioning will stop if the action fails. By default, Fail on Error is not selected for actions created in the Un-deploying, Un-reserving, and Un-initializing stages. ○ Error on Timeout - If selected, provisioning or de-provisioning will stop if the time

Item	Description
	to execute the action exceeds the timeout value, configured below.
Timeout (seconds)	The amount of time to wait for the action to complete. Set this field to zero (0) if you do not want the action to time out. Note that this creates the potential of an action not finishing, which may cause the underlying subscription to remain in the Pending state.

- **Configure Parameters** - Configure the input parameters whose values are passed to the flow or internal action prior to invoking the action. Click the gear icon next to the parameter you want to configure and select **Edit Mapping**. From the Edit Parameter Mapping dialog, edit the Name, Display Name, Description, and Mapping Type of the parameter.

  Select one of the following Mapping Types:

  ○ Not Mapped - The parameter is not mapped to a value.

  ○ Constant Value - Type a value for the parameter.

Lifecycle Action Internal Actions

For more information about lifecycles, see "Lifecycle Actions for Resource Offerings" on page 150

The following internal actions ship with Codar:

Internal Action	Description	Applies To
Build Resource Provider and Pool List	Builds a candidate list of resource providers and associated resource pools that meet the following requirements: • Support the resource offering. • Have an **Availability** of **Enabled**. • The provider's resource pool has sufficient resource capacity. To determine this, you must consider all measurable properties as configured in the **Measurable Properties** tab for the resource offerings, as well as the optional **Multiplier Property Name** field. The resource pool must have enough resource capacity to support all the properties, which requires that each necessary resource type (for example: CPU, Memory, and Storage), based on the measurable properties, be configured on the pool either with a **Resource Availability** of **Unlimited** or **Available**. If **Available**, the difference between **Total Available To CSA** and **Current CSA Utilization** must be sufficient to support the measurable property requirements.	Resource Offering associated with a Service Component
Build Resource Provider List	Builds a candidate list of resource providers that meet the following requirements: • Support the resource offering. • Have an **Availability** of **Enabled**.	Resource Offering associated with a Service Component
Clone Pattern	Clones a service component that is marked as a **Pattern** into one or more non-pattern service components. The number of service components created is determined by the value of the property specified in **Name of the Property for Service Component Count**.	Service Component
Decrease Resource Utilization	Decreases the utilization of one or more resources in a resource pool by the values of the measurable properties configured on a resource offering associated with a service component. **Note:** The Decrease Resource Utilization action cannot be added to a component, component template, or resource offering.This action runs automatically during Un-reserving when Resource Accounting is enabled for a resource offering associated with a service component.	Resource Offering associated with a Service Component
Increase	Increases the utilization of one or more resources in a resource pool by	Resource

Internal Action	Description	Applies To
Resource Utilization	the values of the measurable properties configured on a resource offering associated with a service component. **Note:** The Increase Resource Utilization action cannot be added to a component, component template, or resource offering. This action runs automatically during Reserving when Resource Accounting is enabled for a resource offering associated with a service component.	Offering associated with a Service Component
Log Messages	Writes the user-specified Boolean Input, Integer Input, and String Input property values to the `csa.log` file. You can include this as an action on a Service Component, Resource Offering associated with a service component, or Resource Offering for use in troubleshooting.	Resource Offering Resource Offering associated with a Service Component Service Component
Select Resource Provider	Selects a resource provider from the candidate list that was built by the **Build Resource Provider List** action. The selected resource provider will be available to resource offering actions in the token `RSC_PROVIDER_ID`. The selected provider will, optionally, be written to a property on the associated service component if the `Provider Property Name` input to the action is provided.	Resource Offering associated with a Service Component
Select Resource Provider and Pool	Selects a resource pool and provider from the candidate list that was built by the **Build Resource Provider and Pool List** action. The selected resource provider and pool will be available to resource offering actions in the token `RSC_PROVIDER_ID` and `RSC_POOL_ID`, respectively. The selected pool will, optionally, be written to a property on the associated service component if the `Pool Property Name` input to the action is provided.	Resource Offering associated with a Service Component
Select Resource Provider and Pool from Parent	Selects the resource pool and provider already chosen by a service component's parent service component, as identified by the `Parent Component ID` and `Pool Property Name` properties. The selected resource provider and pool will be available to resource offering actions in the token `RSC_PROVIDER_ID` and `RSC_POOL_ID` respectively. The selected pool will also be written to a `Pool Property Name` property on the associated service component.	Resource Offering associated with a Service Component
Select Resource Provider from	Selects the resource provider already chosen by a service component's parent service component, as identified by the `Parent Component ID` and `Provider Property Name` properties. The selected resource provider will be available to resource offering actions in the token `RSC_`	Resource Offering associated with a

Internal Action	Description	Applies To
Parent	`PROVIDER_ID`. The selected resource provider will also be written to a `Provider Property Name` property on the associated service component.	Service Component

User Operations for Resource Offerings

User operations are actions configured on resource offerings or service, component templates, components in a design that can be invoked by users in the Operations area after a design has been provisioned. For example, a Restart Server user operation can be configured on a server service component or on an associated resource offering, allowing a user to restart a particular provisioned server after the design has been provisioned. User operations can also be invoked on a service instance in the Operations area.

Tasks

- **Navigate to** (the **User Operations** tab) - In the left pane of the **Resource Offerings** area, select the category or provider type associated with the resource offering you want to view. Select the resource offering and then select the **User Operations** tab.

- View a list of user operations for this resource offering.

- **Add Action** - Click **Add**, and enter the information as described in "User Operation Action Selection Wizard" on page 216.

- **Edit Action** - Select the gear icon on the action you want to edit and select **Edit**. Update the information as described in "User Operation Action Selection Wizard" on page 216.

- **View User Operation** - Click the **User Operations** tab to view all user operations associated with a resource offering.

- **Delete User Operation** - Click the gear icon next to the user operation you want to remove, and select **Delete**.

Action Selection Wizard

Use the action selection wizard to create or edit a user operation.

To add Operations Orchestration flows, see the Import Operations Orchestration Flows section in the *Configuration Guide* for more information.

Tasks

- **Select Action** - Search for flows or actions by name (when searching for a flow, the folders searched in the Operations Orchestration library are determined by a property configured in the `csa.properties` file; see the "Action Selection Wizard" property description section in the Configuration Guide for more information) or select the process engine from which to select a flow or action. Then, locate and select the flow or action.

- **Configure Action** - Configure information about the action.

Item	Description
Process Engine	Read-only. The container of the flow or internal action.
Locator	Read-only. The location of the flow or internal action.
Display Name	The name you provide for the action.
Description	The description you provide for the action.
Execution Properties	Specify the following: ○ Fail on Error - During provisioning or de-provisioning, the Fail On Error setting indicates whether provisioning or de-provisioning should continue after an action has failed. For a user operation, the Fail on Error setting is irrelevant because there are no subsequent actions that would run after the user operation. ○ Error on Timeout - If selected, the user operation will be considered to have failed if the time to execute the action exceeds the Timeout value (see below).
Timeout (seconds)	The amount of time to wait for the action to complete. Set this field to zero (0) if you do not want the action to time out.

- **Configure Parameters** - Configure the input parameters whose values are passed to the flow or internal action prior to invoking the action. Click the gear icon next to the parameter you want to configure and select **Edit Mapping**. From the Edit Parameter Mapping dialog, edit the Name, Display Name, Description, and Mapping Type of the parameter.

Select one of the following Mapping Types:

- Not Mapped - The parameter is not mapped to a value.

- Constant Value - Type a value for the parameter.

- Prompt User - Allows the user to enter a value for the parameter when running the user operation. A default value for the parameter is required.

- Prompt User List - Allows the user to select a value for a parameter by selecting from a list of items when running the user operation.

List Items	
Switch to Static Entry (default)	Enter the display name and value for a list item. After entering a name or value, input boxes for a new list item automatically display.
Switch to Dynamic Entry	- Select a script from the List Items. **HTTP Parameters** - Specify the HTTP parameters to use when executing the script. The required input parameters are dependent on the script that has been selected; contact the script developer if necessary to determine the necessary input parameters. **Refresh Data** - Click the Refresh Data icon ( ) to run the script and display the results.

Custom Resource Offering Properties

For more information about resource offerings, see "Resource Offerings".

Custom properties for resource offerings are user-defined properties that are used for communicating properties and values to providers to customize the functional service they provide. The custom properties can be used to provide any information that is expected by an Operations Orchestration flow. For example, you can use custom properties for a resource offering to provide a specific VM_TEMPLATE_ NAME value to an instance of VMware vCenter.

See the following related topics:

- "View Custom Resource Offering Properties" on the next page

- "Create Custom Properties" on page 163

- "Delete resource-offering property" on page 165

- "Edit Properties of a Resource Offering" on page 166

View Custom Resource Offering Properties

For more information about custom offering properties, see "Custom Resource Offering Properties" on the previous page.

To view custom properties for a resource offering

1. In the **Offerings** tab, click the **Display Name** of the offering whose custom properties you want to view.

2. In the **Properties** tab, review the custom properties created for the offering.

Create Custom Properties

To create a custom property for an offering, in the **Offerings** tab, select the offering. In the offering's **Properties** tab, click **Create**. Enter the following information in the **Create Property** wizard:

Item	Description
Property Type	Select one of the following: • **Boolean** - A property whose value is true or false. • **List** - A property whose value is a list of String values. • **Integer** - A property whose value is a positive or negative whole number or zero. • **String** - A property whose value is a sequence of characters. You cannot edit this value after the property has been created.
Property Details	*For Boolean properties:* • **Name** - A unique name for the property. • **Display Name** - The display name for the property. • **Description** - A description of the property. • **Property Value** - Select **true** or **false**. *For List properties:* • **Name** - A unique name for the property. • **Display Name** - The display name for the property. • **Description** - A description of the property. • **Value Entry Method**: - Select **Manual Entry** to provide **Specified Values** for this field, as described below. • **Specified Values** - Click the **Add Value** icon to add a new value, or click the **Remove Selected Value(s)** icon to remove a selected value. *For Integer properties*: • **Name** - A unique name for the property. • **Display Name** - The display name for the property. • **Description** - A description of the property. • **Property Value** - Select or type a positive or negative whole

Item	Description
	number or zero. If you enter a decimal number, the value will be truncated to the nearest integer. The maximum allowed integer value is 2147483647 and the minimum is -2147483648; if you enter a value outside these bounds the value will be automatically converted to the closest maximum or minimum value. *For String properties:* • **Name** - A unique name for the property. • **Display Name** - The display name for the property. • **Description** - A description of the property. • **Property Value** - Type a string of characters. • **Confidential Data** - Select this box to mask the values so that they cannot be read in the user interface; no encryption of the value is performed.

Delete resource-offering property

For more information about resource offerings, see "Resource Offerings".

The properties of a resource offering cannot be deleted if the resource offering is linked to a service component that is part of a service design, where the service design is in an upgrade path, is upgradable from a previous version of the design, and the property existed in that previous version.

The properties of a resource offering can be deleted if the resource offering is not linked to a service component, if the resource offering is linked to a service component of a design that is not upgradable from a previous version of the design, or the properties are newly added to the current version of the design.

To delete a property of a resource offering, in the resource offering's summary page click the **Properties** tab. Click the gear icon for the property to be edited and choose **Delete**, then confirm your choice.

For descriptions of certain properties, see "Create a Resource Offering" on page 144.

Edit Properties of a Resource Offering

For more information about resource offerings, see "Resource Offerings".

To edit properties of a resource offering, in the resource offering's summary page click the **Properties** tab. Click the gear icon for the property to be edited and choose **Edit**. Edit the property and click **Save**.

> **Note:** You cannot edit the Type or Name of an existing property.

For descriptions of certain properties, see "Create a Resource Offering" on page 144.

Resource Offering Categories

Resource offering categories allow you to classify resource offerings for improved filtering and identification. CSA includes some predefined categories out-of-the-box (shown below) or you can create your own. A category is associated with a resource offering and can be used when assigning resource offerings to service designs.

Service components within a service design can accept offering assignments only for those offerings with a category that is supported by the service component type. For example, a **Server** service component can be associated with offerings from a category of *Compute* (among others), while a

Software Application Service component can be associated with offerings with a category of
Application.

Resource categories are available by selecting **By Category** from the list in the left navigation pane.

Resource Offering Categories

Resource offering categories allow you to classify resource offerings for improved filtering and identification. CSA includes some predefined categories out-of-the-box (shown below) or you can create your own. A category is associated with a resource offering and can be used when assigning resource offerings to service designs.

Service components within a service design can accept offering assignments only for those offerings with a category that is supported by the service component type. For example, a **Server** service component can be associated with offerings from a category of *Compute* (among others), while a

Software Application Service component can be associated with offerings with a category of *Application*.

Resource categories are available by selecting **By Category** from the list in the left navigation pane.

Manage Resource Category

For more information about categories, see "Resource Offering Categories" on page 169.

In the resource offering's landing page, click the gear and chose **Manage Resource Category**. Use the add, edit, or delete a icons to add, edit, or delete the category.

This topic contains the following sections:

- "Create a resource category" below

- "Edit a resource category" below

- "Delete a resource category" on the next page

Create a resource category

To create a resource category, select **By Category** from the dropdown, then click the gear icon and select **Manage Resource Category**. Click the add icon, add the category properties, and then click **Create**.

Resource category properties:

Item	Description
Display Name	The display name you provide for the category.
Description	The description you provide for the category.
Image	An image that displays for the category. Click **Change Image**. Choose the image you want, and click **Select**. Click **Upload Image** to add your own image. Supported file extensions include `.jpg`, `.jpeg`, `.gif`, and `.png`. The recommended image size is 256 by 256 pixels, and the image will be scaled to the appropriate size. The images are stored in the `%CSA_HOME%\jboss-as\standalone\deployments\csa.war\images\library` folder of the Codar server.

Edit a resource category

To edit a resource category, choose to filter **By Category**, then select the category to edit. Click the gear icon and choose **Manage Resource Category**. Click the edit icon, edit the category properties, click **Save**, and then click **Done**.

Delete a resource category

Categories cannot be deleted:

- If offerings are using them.

- If they are out-of-the-box.

To delete a category, choose to filter **By Category**, then select the category to delete. Click the gear icon and choose **Manage Resource Category**. Click the delete icon, confirm your deletion, then click **Done**.

View Component Palettes

For more information about components, see Components (Sequenced Designs).

To view component palettes

In the left pane of the **Sequenced Components** area, view the available palettes. The list of component types contained in the selected palette displays in the right pane.

Use the following icons and features to navigate and perform tasks in the **Sequenced Components** area:

Item	Description
🔒	Indicates locked items, which cannot be edited or deleted. However, you can create component types and component templates from a locked component type.
Search Box	Enter search text to filter the results.

Create a Component Palette

For more information about components, see Components.

To create a component palette

1. Click the gear icon and select **Manage Component Palettes**.

2. Click the **Create Component Palette** icon.

3. Provide information as described in the table below and then click **Create**.

4. Click **Done**.

Field	Description
Name	A unique name for the component palette. No white space is permitted; all letters are changed to upper case for this value. This field cannot be modified after a component palette has been created.
Display Name	The name that displays, for the component palette.
Description	The description for the component palette.
Image	An image that displays for the component palette. Click **Change Image**. Choose the image you want, and click **Select**. Click **Upload** to add your own image. Supported file extensions include `.jpg`, `.jpeg`, `.gif`, and `.png`. The recommended image size is 256 by 256 pixels, and the image will be scaled to the appropriate size. The images are stored in the `%CSA_HOME%\jboss-as\standalone\deployments\csa.war\images\library` folder of the CSA server.

Edit a Component Palette

For more information about components, see Components.

You cannot edit a locked component palette, as designated by the **Lock** icon.

To edit a component palette

1. Click the gear icon and select **Manage Component Palettes**.

2. In the Component Palettes dialog, select a component you want to edit.

3. Click the **Edit Component Palette** icon.

4. Provide the desired information, and click **Save**. For information about the specific fields, see "Create a Component Palette" on the previous page.

5. Click **Save** and then click **Done**.

Delete a Component Palette

For more information about components, see Components.

When you delete a palette, all of its component types (including all component properties, constraints, and templates) are deleted. The deletion can take a few minutes depending on the number of component types in the palette. Deletion of the component palette cannot be undone.

You cannot delete a component palette in the following cases:

- If the palette is locked, as designated by the **Lock** icon.

- If any component type contained in the palette is being used in a service design.

- If any other component palette depends on the palette. That is, if another palette contains a component type that is based on a component type from the palette.

To delete a component palette

1. Click the gear icon and select **Manage Component Palettes**.

2. In the Component Palettes dialog, select a component you want to delete.

1. Click the **Delete Selected Component Palette** icon.

2. Click **Yes** in the confirmation dialog.

3. Click **Done**.

Import and Export a Component Palette

For more information about components, see Components.

Concepts

In this topic, artifacts refer to a component palette and its associated component types, templates, component type constraints, and resource offerings.

You can import and export many of the artifacts that provide the basis for cloud automation. The export operation provides the ability to preserve the selected artifacts so they can be used to replicate the services on another system or to restore the artifacts. These exported archive files are preserved in an industry-standard zip archive file format.

The import operation provides the ability to install and replace artifacts. See the *Tasks* section below for more information.

Process Information

- **Resource offerings** - If a component template in a component palette is associated with a resource offering, when the component palette is exported, resource offering XML files are included in the component palette archive. Therefore, when importing this archive, the resource offerings are also imported.

- **Import operation** - The import operation is destructive to existing data (unlike the import operations for service designs and resource offerings for which there are options to only import new artifacts without overwriting existing ones). If an imported artifact exists on the target system, the existing artifact on the target system is overwritten. For example, if the imported component palette contains a resource offering that exists on the target system, the existing resource offering on the target system is overwritten by the resource offering from the archive.

  Select the preview button to view prospective results for the import process.

- **Import process integrity** - The import process for component palettes always ensures system integrity with respect to component type derivation and property propagation. For example, say you have modified properties of a component type in a palette called "Palette A," and there are component types in other palettes that are derived from this component type. When an update of "Palette A" occurs during import, changes that have been made to the component types are propagated to component types in other palettes.

- **Palette dependencies** - Import multiple, dependent palettes in the order of their dependencies if these palettes exist on the target system.

- **Circular dependencies** - Circular dependencies between palettes are not allowed. For example, Palette A cannot have components that are derived from Palette B while, at the same time, Palette B has components that are derived from Palette A.

- **Flows** - During an import, update, or preview operation, if required dependencies do not exist on the Operations Orchestration system, an error message identifies these missing dependencies (dependencies such as flows). The content pack that contain these flows must have been deployed to the Operations Orchestration system prior to importing these artifacts. The flows must also have identical signatures and identical paths as the flows on the system from which the artifact was exported.

  During import, flow signature-related information is verified in or added to the Codar database (flow signatures are used during the creation of an artifact and when adding a resource synchronization action or an external approval type). This information is resolved by **name** which corresponds to the full path to the Operations Orchestration flow (for example, `/Library/CSA Content Pack/CSA3.2/Providers/Infrastructure/vCenter/vCenter Clone Server/Actions/vCenter Simple Compute - Deploy`). For information about how to deploy Operations Orchestration content packs, see the *Central User Guide*.

- **Archive content** - The archive (`.zip`) file can only reference files or content contained within the `.zip` file itself, or that are already contained in the `csa.war` file.

- **Image files** - All images in the archive files must end in one of the following suffix values (for information on adding additional suffix values, see the *Configuration Guide*):

  `jpg|jpeg|jpe|jfif|svg|tif|tiff|ras|cmx|ico|pnm|pbm|pgm|ppm|rgb|xbm|xpm|xwd|png`

- **Back up your data** - Create a backup of your system or data. Before proceeding, be sure to create a backup of any artifacts you may be affecting by using the export operation to save an archive zip file.

Tasks

Import and export tasks are completed from the **Manage Component Palettes** dialog.

- **Import** - Imports new artifacts and overwrites existing artifacts. New artifacts are created if they do not exist on the target system. Identical artifacts that exist on the target system are overwritten with changes from the archive.

> **Note:** The import operation identifies artifacts by name to determine if the artifact exists on the target system.

- **Preview** - Generate a report of prospective results for the import process, including information about the artifacts and their status.

- **View Detailed Report** - From the Import Summary dialog, select this link to see a summary and details of the import process, including information about the artifacts and their status.

- **Export** - Create a content archive (.zip) file. The content archive contains XML documents for the artifacts you are exporting, as well as icons for customizing the artifacts, and the Manifest XML document, which contains meta-information about the archive files.

  The component palettes are packaged in an archive file whose name is:

  `COMPONENT_PALETTE_<component_palette_display_name>_<component_palette_id>.zip`

For more information about importing and exporting artifacts, see the *Cloud Service Automation Content Archive Tool* document.

View Component Types

For more information about components, see Components.

To view component types

In the left pane of the **Sequenced Components** area, select the palette that contains the component type you want to view. The list of component types contained in the selected palette displays in the right pane.

Use the following icons and features to navigate and perform tasks in the **Sequenced Components** area:

Item	Description
🔒	Indicates locked items, which cannot be edited or deleted. However, you can create component types and component templates from a locked component type.
Search Box	Enter search text to filter the results.
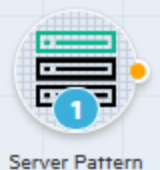
Create a Component Type

For more information about component types, see Components (Sequenced Designs).

To create a component type

1. In the left pane of the **Sequenced Components** area, select the component palette in which you want to create a component type.

2. Click **Create Component Type**.

3. Provide the following information and then click **Create.**

Item	Description
Base Component Type	Select the component palette from which you want to select a base component type, and then select a base component type from which the new component type will inherit properties and constraints. Component types are not required to have a **Base Component Type**. The lists display only component palettes and component types that are compatible with (that is, they do not create circular dependencies with) the palette in which you are creating the component type. This field cannot be modified after a base component type has been selected.
Name	A name for the component type. The component type name must be unique within a component palette. No white space is permitted; all letters are changed to upper case for this value. This field cannot be modified after a component type has been created.
Display Name	The name that displays for the component type.
Description	The description you provide for the component type.
Image	An image that displays for the component type. Click **Change Image**. Choose the image you want, and click **Select**. Click **Upload** to add your own image. Supported file extensions include `.jpg`, `.jpeg`, `.gif`, and `.png`. The recommended image size is 256 by 256 pixels, and the image will be scaled to the appropriate size. The images are stored in the `%CSA_HOME%\jboss-as\standalone\deployments\csa.war\images\library` folder of the Codar server.
Default Settings	Select the following items as needed. These items specify the initial default values for service components: • **Consumer Visible** - Select this box to specify that, by default, the service component is visible in the Marketplace Portal. Deselect it to specify that the service component will not be visible. Properties and lifecycle actions that have been configured as consumer visible will

Item	Description
	only display in the Marketplace Portal if the associated service component itself is configured to be visible. The visibility of a service component does not affect visibility of its child service components, and it is not necessary for a parent of a service component to be visible for the child component to be visible in the Marketplace Portal. **Note:** Marketplace Portal is available only when the CSA license is applied. • **Pattern** - Select this box to mark the service component as a pattern by default. This indicates that the service component will not be automatically processed by the lifecycle engine. When you create a pattern service component, you must also create a Clone Pattern action in the parent service component. If the Clone Pattern action is not created in the parent service component, a validation error is displayed in the designer. For example, the out-of-the-box component type, Server, is typically marked as a pattern, and its parent service component, Server Group, includes the Clone Pattern action. ○ You can identify a service component that is marked as a pattern because its icon appears with a grid-like background, as shown below. Server Pattern

View Component Type Details

For more information about components, see Components.

To view the details of a component type

1. In the left pane of the **Sequenced Components** area, select the component palette that contains the component type whose details you want to view.

2. Click the component type whose details you want to view.

3. In the **Overview** tab, review details of the component type. For descriptions of the specific properties, see the topic "Create a Component Type" on page 181.

See the following related topics:

- "View Component Type Properties" on page 186

- "View Constraints of a Component Type" on page 192

- "View Component Templates" on page 196

Edit a Component Type

For more information about components, see Components.

You cannot edit a component type if it part of a locked palette or if the component type is locked as a system critical object. In these cases, the **Edit** button is disabled in the **Overview** tab.

To edit a component type

1. In the left pane of the **Sequenced Components** area, select the component palette that contains the component type you want to edit.

2. Click the component type you want to edit.

3. In the **Overview** tab, click the gear icon and select **Edit**.

4. Edit the details of the component type. For descriptions of the specific fields, see the topic "Create a Component Type" on page 181.

5. Click **Save**.

Delete a Component Type

For more information about components, see Components.

You cannot delete a component type in the following cases:

- If it is part of a locked palette or if the component type is locked as a system critical object. In these cases, the **Delete** link is disabled.

- If another component type is based on (derived from) the component type.

- If service components in service designs are based on this type.

To delete a component type

1. In the left pane of the **Sequenced Components** area, select the component palette that contains the component type you want to delete.

2. Click the component type you want to delete.

3. Click the gear icon and select **Delete**.

4. Click **Yes** in the confirmation dialog.

View Component Type Properties

For more information about components, see Components.

Properties for component types provide a base set of attributes that can be used and edited when creating service components in a service design. The value defined for a component type property is the default value exposed in the service design unless the service design uses a template, in which case, the template's property value is set as the default value in the service design.

To view the properties of a component type

1. In the left pane of the **Sequenced Components** area, select the component palette that contains the component type whose properties you want to view.

2. Click the component type whose properties you want to view.

3. Select the **Properties** tab.

4. In the **Properties** tab, the properties are grouped into two lists:
   - **Defined Properties** - Properties created directly on this component type. You can delete, edit, and update defined properties and their values, unless the component is locked. In which case, only the property value can be set and the **Delete** action will be disabled for the property that is part of the locked component type.

   - **Inherited Properties** - Properties inherited from base parent and grandparent component types. Inherited properties cannot be deleted, and you can edit only the following for inherited properties:
     - Property values
     - **Consumer Visible** property
     - **Confidential Data** property
     - **Resource Type and Unit** property (for Integer property types only)

5. For more information about property details, see "Create Component Type Properties" on the next page.

See the following related topics:

- "View Component Type Details" on page 183.

- "View Constraints of a Component Type" on page 192

- "View Component Templates" on page 196

Create Component Type Properties

For more information about components, see Components. For more information about custom service components, see "Service Component Properties" on page 123.

> **Caution:** When you create a property for a component type that has derived subtype components or existing component templates, the property is added to all associated subtypes and templates. Because service designs are considered a snapshot in time of the components, and to preserve the integrity of service designs that are provisioned, the propagation of component type properties does not affect service designs. You must manually update any service design that uses the component type, its subtypes, or its templates.

To create component type properties

1. In the left pane of the **Sequenced Components** area, select the component palette that contains the component type whose properties you want to create.

2. Click the component type whose properties you want to create.

3. Select the **Properties** tab.

4. Click **Create** to open the Create Property dialog.

5. Select one of the following types:

   - **Boolean** - A property whose value is true or false.

   - **Integer** - A property whose value is a positive or negative whole number (or zero).

   - **List** - A property whose value is a list of string values.

   - **String** - A property whose value is a sequence of characters.

   You cannot change the type once the property has been created.

4. Depending on the **Type** selected above, provide the following property information and then click **Create**:

Type	Property Information
Boolean	○ **Name** - A unique name for the property. ○ **Display Name** - The label that will be displayed for the property. ○ **Description** - A description of the property. ○ **Property Value** - Select either **True** or **False**. ○ **Consumer Visible** - Select this box to indicate that this property will be made visible in the Marketplace Portal.
Integer	○ **Name** - A unique name for the property. ○ **Display Name** - The label that will be displayed for the property. ○ **Description** - A description of the property. ○ **Resource Type and Unit for a Measurable Property** - Select the resource type and unit, which are used to influence provider and pool selection and resource accounting. You must set this to a value other than **None** to make this property measurable and available for configuration on a resource offering. You can also specify the resource type and unit simply to clarify the intent of the property. Note that when creating a **Source Binding** for a service component, a best practice is that both the service component property you are creating and the service component property you are binding to will have the same **Resource Type and Unit** value. When you change this value for a component type that has derived subtype components or existing component templates, the changes you make apply to all associated subtypes and templates. ○ **Property Value** - Type a positive or negative whole number or zero. If you enter a decimal number, the value will be truncated to the nearest integer. The maximum allowed integer value is 2147483647 and the minimum is -2147483648. If you enter a value outside these bounds the value will be automatically converted to the closest maximum or minimum value. ○ **Consumer Visible** - Select this box to indicate that this property will be made visible in the Marketplace Portal.
List	○ **Name** - A unique name for the property. ○ **Display Name** - The label that will be displayed for the property. ○ **Description** - A description of the property. ○ **Property Values** • Click the **Add Item** icon to add a new list item, and provide the following information: • **Display Name**- The label that will be displayed for the list item. • **Value** - The value of the list item. Values must be unique with a list. • **Description** A description of the list item.

Type	Property Information
	● Click **Add** to add the list item. ● Click the **Edit Item** icon to edit a selected list item. ● Click the **Remove Item** icon to remove a selected list item. ○ **Consumer Visible** - Select this box to indicate that this property will be made visible in the Marketplace Portal.
String	○ **Name** - A unique name for the property. ○ **Display Name** - The label that will be displayed for the property. ○ **Description** - A description of the property. ○ **Property Value** - Type the value of the property. The value must be in string format. Click the gear icon to select a property value token. See **Tokens** for a description of the tokens. ○ **Consumer Visible** - Select this box to indicate that this property will be made visible in the Marketplace Portal. ○ **Confidential Data** - Select this box to mask the values so that they cannot be read in the Marketplace Portal; no encryption of the value is performed.

Note: Marketplace Portal is available only when the CSA license is applied.

Edit Component Type Properties

For more information about components, see Components.

For inherited properties, you cannot edit the following fields:

- Type

- Name

- Display Name

- Description

- Confidential Data

To edit component type properties

1. In the left pane of the **Sequenced Components** area, select the component palette that contains the component type whose properties you want to edit.

2. Click the component type whose properties you want to edit.

3. Select the **Properties** tab.

4. Click the gear icon for the property and select **Edit**.

5. In the Edit Component Type dialog, change properties of the component type. For descriptions of certain fields, see "Create Component Type Properties" on page 187.

6. Click **Save**.

Delete Component Type Properties

For more information about components, see Components.

> **Caution:** When you delete a property from a component type that has derived subtype components or existing component templates, the deletion applies to all associated subtypes and templates. Because service designs are considered a snapshot in time of the components, and to preserve the integrity of service designs that are provisioned, the propagation of component type properties does not affect service designs. You must manually update any service design that uses the component type, its subtypes, or its templates.

You cannot delete a component type property in the following cases:

- If the property is part of the original locked component type. In this case, in the component property **Defined Properties** list, the **Delete** action will be disabled. A locked property can also be identified when in the edit view. The locked icon will appear in the lower left hand corner of the edit dialog.

- If the property is inherited from another component type.

To delete component type properties

1. In the left pane of the **Sequenced Components** area, select the component palette that contains the component type whose properties you want to delete.

2. Click the component type whose properties you want to delete.

3. Select the **Properties** tab.

4. In the Properties tab, click the gear icon for the property and select **Delete**.

5. Click **Yes** in the confirmation dialog.

View Constraints of a Component Type

For more information about components, see Components.

Service Designers can configure two kinds of constraints for a component type: component type constraints and resource category constraints. These constraints apply to service components within a service design that are created from this component type.

Component type constraints limit the types of service components that can be connected to components of this type within a service design. For example, a service component of type Server may be allowed connections only to components of types Application Layer, Network Connection, Software Component, and Storage Volume.

Resource category constraints limit the categories of resource offerings that can be bound to service components of this type. For example, a Server service component may be allowed only to resource offerings associated with service components that are assigned the categories Application, Compliance, Compute, Configuration Management, Monitoring, Network, Service Usage, and Storage.

A component type can establish constraints in the following ways:

- *Defined constraints* - specified directly on component types.
- *Inherited constraints* - inherited by component types from the hierarchy of types they are derived from. In the list of constraints, the path of inheritance is listed next to the name of each component type.

To view the constraints of a component type

1. In the left pane of the **Sequenced Components** area, select the component palette that contains the component type whose constraints you want to view.

2. Click the component type whose constraints you want to view.

3. Select the **Constraints** tab.

4. Select the type of constraint you want to view: Component Types or Resource Categories.

5. Review the list of constraints.

See the following related topics:

- "View Component Type Details" on page 183
- "View Component Type Properties" on page 186

Use

View Constraints of a Component Type

-

Add a Constraint

For more information about components, see Components.

For information about constraints, see "View Constraints of a Component Type" on page 192.

To add constraint

1. In the left pane of the **Sequenced Components** area, select the component palette that contains the component type to which you want to add a constraint.

2. Click the component type to which you want to add a constraint.

3. Select the **Constraints** tab.

4. Select the type of constraint you want to add: Component Types or Resource Categories.

5. Click **Add**.

6. In the Add Constraint dialog, make your selection and then click **Add**.

Remove a Constraint

For more information about components, see Components.

For information about constraints, see "View Constraints of a Component Type" on page 192.

You cannot remove a constraint from a component type if the component type is locked. In addition, you cannot remove a constraint that is delivered by Codar; however, you can add constraints to the Codar delivered component types, and then you can remove constraints that you added.

To remove a constraint

1. In the left pane of the **Sequenced Components** area, select the component palette that contains the component type from which you want to remove a constraint.

2. Click the component type from which you want to remove a constraint.

3. Select the **Constraints** tab.

4. Select the type of constraint you want to delete: Component Types or Resource Categories.

5. Click the gear icon for the constraint and select **Delete**.

6. Click **Yes** in the confirmation dialog.

View Component Templates

For more information about components, see Components.

A component template is a specialized version of a component type and is used to simplify service design creation. Component templates include customized settings for the properties, lifecycle actions, and resource offerings that provide initial settings for the service design.

To view the templates that are based on a component type

1. In the left pane of the **Sequenced Components** area, select the component palette that contains the component type whose templates you want to view.

2. Click the component type whose templates you want to view.

3. Select the **Templates** tab.

4. The **Templates** tab displays the list of templates created from the selected component type.

You can also see component template information in the following tabs:

- **Overview** tab - See "View Component Template Details" on page 201

- **Properties** tab - See "View Component Template Properties" on page 202.

- **Lifecycle** tab - See "Lifecycle Actions for Component Templates" on page 207

- **Resource Offerings** tab - See "Associate Resource Offerings with Service Components" on page 105.

Use the following icons and features to navigate and perform tasks in the **Sequenced Components** area:

Item	Description
🔒	Indicates locked items, which cannot be edited or deleted. However, you can create component types and component templates from a locked component type.
Search Box	Enter search text to filter the results.

Create Component Templates

For more information about components, see Components.

To create component templates

1. In the left pane of the **Sequenced Components** area, select the component palette that contains the component type from which you want to create a template.

2. Click the component type on which you want to base the template.

3. Select the **Templates** tab and click **Create**.

4. In the **Create Component Template** dialog, enter the following information and click **Create**.

Item	Description
Base Component Type	The component type from which this template is created; this field cannot be edited.
Display Name	The name you that is displayed for the component template.
Description	The description you provide for the component template.
Image	An image that displays for the component type. Click **Change Image**, as desired. Choose the image you want, and click Select. Click Upload Image to add your own image. Supported file extensions include .jpg, .jpeg, .gif, and .png. The recommended image size is 256 by 256 pixels, and the image will be scaled to the appropriate size.
Consumer Visible	Select this box to specify that the service component is visible in the Marketplace Portal. De-select it to specify that the service component will not be visible. **Note:** Marketplace Portal is available only when the CSA license is applied.
Pattern	Select this box to mark the service component as a pattern by default. When you create a pattern service component, you must also create a Clone Pattern action in the parent service component. For example, the out-of-the-box component type, Server, is typically marked as a pattern, and its parent service component, Server Group, includes the Clone Pattern action.

Edit Component Templates

For more information about components, see Components.

To edit component type templates

1. In the left pane of the **Sequenced Components** area, select the component palette that contains the component type whose template you want to edit.

2. Click the component type that contains the template you want to edit.

3. Select the **Templates** tab.

4. Click the template you want to edit.

5. Click the gear icon and select **Edit**.

6. In the Edit Component Template dialog, make your changes and click **Save**. For information about certain fields, see "Create Component Templates" on the previous page.

Copy Component Templates

For more information about components, see Components.

To copy component type templates

1. In the left pane of the **Sequenced Components** area, select the component palette that contains the component type whose template you want to copy.

2. Click the component type that contains the template you want to copy.

3. Select the **Templates** tab.

4. Click the template you want to copy.

5. Click the gear icon and select **Save As**.

6. In the Copy Component Template dialog, make your changes and click **Save**. For information about certain fields, see "Create Component Templates" on page 197.

Delete Component Templates

For more information about components, see Components.

To delete component type templates

1. In the left pane of the **Sequenced Components** area, select the component palette that contains the component type whose templates you want to delete.

2. Click the component type whose templates you want to delete.

3. Select the **Templates** tab.

4. Select the template you want to delete

5. Click the gear icon and select **Delete**.

6. Click **Yes** in the confirmation dialog.

View Component Template Details

For more information about components, see Components (Sequenced Designs).

To view the details of a component template

1. In the left pane of the **Sequenced Components** area, select the component palette that contains the component type whose template details you want to view.

2. Click the component type whose template details you want to view.

3. Select the **Templates** tab.

4. In the templates list, click the template whose details you want to view.

5. In the **Overview** tab, see the details of the component template. For descriptions of certain details, see "Create Component Templates" on page 197.

See the following related topics:

- "Create Component Template Properties" on page 203

- "Lifecycle Actions for Component Templates" on page 207

View Component Template Properties

For more information about components, see Components.

To view the properties of a component template

1. In the left pane of the **Sequenced Components** area, select the component palette that contains the component type whose template properties you want to view.

2. Click the component type whose template properties you want to view.

3. Select the **Templates** tab.

4. In the template list, click the component template whose properties you want to view.

5. In the **Properties** tab, the list includes defined and inherited properties:
   - **Defined Properties** - Properties created directly on this component template. You can delete, edit, and update defined properties and their values. New component templates have no defined properties, unless the component type is locked. If the component type is locked, only the property value can be set in the template's properties.

   - **Inherited Properties** - Indicated by the **Inherited** label. Properties inherited from the base component type hierarchy. Identified in this list by the **Inherited** label. You can edit only the values of inherited properties.

6. For more information about property details, see "Create Component Template Properties" on the next page.

See the following related topics:

- "View Component Template Details" on the previous page

- "Lifecycle Actions for Component Templates" on page 207

Create Component Template Properties

For more information about components, see Components.

To create component template properties

1. In the left pane of the **Sequenced Components** area, select the component palette that contains the component type whose template properties you want to create.

2. Click the component type whose template properties you want to create.

3. Select the **Templates** tab.

4. In the component templates list, click the component template whose properties you want to create.

5. Click **Create**.

6. In the Create Property dialog, enter the following information and then click **Create**.

Item	Description
Type	Select one of the following: • **Boolean** - A property whose value is true or false. • **List** - A property whose value is a list of String values. • **Integer** - A property whose value is a positive or negative whole number (or zero). • **String** - A property whose value is a sequence of characters.
Depending on the **Type** selected above, provide the following information:	
Boolean	• **Name** - A unique name for the property. • **Display Name** - The label that will be displayed for the property. • **Description** - A description of the property. • **Property Value** - Select either **true** or **false**. • **Consumer Visible** - Select this box to indicate that this property will be made visible in the Marketplace Portal.
Integer	• **Name** - A unique name for the property. • **Display Name** - The label that will be displayed for the property. • **Description** - A description of the property. • **Resource Type and Unit for a Measurable Property** - Select the resource

Item	Description
	type and unit, which are used to influence provider and pool selection and resource accounting. You must set this to a value other than **None** to make this property measurable and available for configuration on a resource offering. You can also specify the resource type and unit simply to clarify the intent of the property. Note that when creating a **Source Binding** for a service component, a best practice is that both the service component property you are creating and the service component property you are binding to will have the same **Resource Type and Unit** value. • **Property Value** - Type the value of the property. The value must be an integer. • **Consumer Visible** - Select this box to indicate that this property will be made visible in the Marketplace Portal.

List	• **Name** - A unique name for the property.
	• **Display Name** - The label that will be displayed for the property.
	• **Description** - A description of the property.
	• **Property Value**
	  ○ Click the **Add Value** icon to add a new list item, and provide the following information:
	    • **Display Name**- The label that will be displayed for the list item.
	    • **Value** - The value of the list item.
	    • **Description** A description of the list item.
	    • Click **Create** to add the list item.
	  ○ Click the **Edit** icon to edit the selected list item.
	  ○ Click the **Remove** icon to remove a selected list item.
	• **Consumer Visible** - Select this box to indicate that this property will be made visible in the Marketplace Portal.

String	• **Name** - A unique name for the property. • **Display Name** - The label that will be displayed for the property. • **Description** - A description of the property. • **Property Value** - Type the value of the property. The value must be in string format. Click the gear icon to select a property value token. • **Consumer Visible** - Select this box to indicate that this property will be made visible in the Marketplace Portal. • **Confidential data** - Select this box to mask the values so that they cannot be read in the Marketplace Portal; no encryption of the value is performed.

Note: Marketplace Portal is available only when the CSA license is applied.

Edit Component Template Properties

For more information about components, see Components.

For inherited properties, you cannot edit the following fields:

- Type

- Name

- Display Name

- Description

- Confidential Data

To edit component template properties

1. In the left pane of the **Sequenced Components** area, select the component palette that contains the component type whose template properties you want to edit.

2. Click the component type whose template properties you want to edit.

3. Select the **Templates** tab.

4. In the component templates list, click the component template whose properties you want to edit.

5. Click the gear icon for the property and select **Edit**.

6. In the Edit Properties dialog, make your changes to the component template properties. For descriptions of certain fields, see "Create Component Template Properties" on page 203.

Delete Component Template Properties

For more information about components, see Components.

You cannot delete a component template property in the following cases:

- If the property is locked, as designated by the **Lock** icon.

- If the property is inherited.

To delete component template properties

1. In the left pane of the **Sequenced Components** area, select the component palette that contains the component type whose template properties you want to delete.

2. Click the component type whose template properties you want to delete.

3. Select the **Templates** tab.

4. In the templates list, click the template whose properties you want to delete.

5. Click the gear icon for the property and select **Delete**.

6. Click **Yes** in the confirmation dialog.

Lifecycle Actions for Component Templates

For more information about component templates, see "View Component Templates" on page 196

Concepts

What is a component template lifecycle?

A component template lifecycle is a collection of actions defined for a component template. The Lifecycle area allows you to specify the actions that are needed to provision, update, and deprovision component templates. Use the action selection wizard to add a lifecycle action to a component template from one of the available process engines.

What is a lifecycle action?

A lifecycle action is a function that is either run automatically at a specified lifecycle stage or phase, or that is exposed to the user. Most lifecycle actions correspond to OO flows, which contain the logic for executing the function. Codar also includes actions that run internal to Codar. Many Codar internal actions relate to provider and pool selection and resource accounting. Actions include input parameters that provide configuration information to the function.

What are lifecycle stages?

A lifecycle stage represents a step within the Codar service lifecycle. There are three categories of lifecycle stages: Provisioning, Operational, and De-provisioning.

The Provisioning lifecycle stages are:

- **Initializing** - Initializing is the first stage that is processed during provisioning. Use the Initializing stage for actions that perform any type of initialization that is required before proceeding to the Reserving stage, such as actions that perform input validation or create change request records.

- **Reserving** - Reserving is the second stage that is processed during provisioning, after the Initializing stage. When a resource offering is associated to a service component, any configured provider selection and pool selection actions run during the Reserving stage. If resource accounting is enabled, actions to reserve resources from a pool run during the Reserving stage. In general, any resources needed during provisioning, such as storage or networking, can be reserved during the Reserving stage.

- **Deploying** - Deploying is the final stage that is processed during provisioning, after the Initializing and Reserving stages, and is the most common stage in which provisioning actions are defined. Define actions in the Initializing and Reserving stages to prepare the environment for provisioning.

Define actions in the Deploying stage to perform actual provisioning. For example, you can define actions that deploy a VM or an application in the Deploying stage.

The De-provisioning lifecycle stages are:

- **Un-deploying** - Un-deploying is the first stage that is processed during de-provisioning and is the most common stage in which de-provisioning actions are defined. Use the Un-deploying stage to undo or to finalize the actions that took place during the Deploying stage. For example, if actions in the Deploying stage deploy a VM, define actions in the Un-deploying stage to un-deploy the VM.

- **Un-reserving** - Un-reserving is the second stage that is processed during de-provisioning, after the Un-deploying stage. Use the Un-reserving stage to undo or to finalize the actions that took place during the Reserving stage. For example, if resource accounting is enabled, actions to return resources to the pool run during the Un-reserving stage.

- **Un-initializing** - Un-initializing is the final stage that is processed during de-provisioning, after the Un-deploying and Un-reserving stages. Use the Un-initializing stage to undo or to finalize the actions that took place during the Initializing stage. For example, if actions in the Initializing stage create a change request, define actions in the Un-initializing stage to finalize that change request.

What are lifecycle phases?

A lifecycle phase is a further refinement of a lifecycle stage. When defining a lifecycle action at a lifecycle stage, you can also specify the phase for the action. The phases are:

- **Before** - Actions configured in the Before phase on a service component run before all actions on child service components in a design and before all actions on any associated resource offerings. Actions configured in the Before phase on a resource offering run after all of the Before actions on the associated service component have completed.

- **During** - Actions configured in the During phase on a service component run after all of the actions have completed on child service components in a design but before any of the During actions run on any associated resource offerings. Actions configured in the During phase on a resource offering run after all of the During actions on the associated service component have completed.

- **After** - Actions configured in the After phase on a service component run after all actions have completed on child service components and after all of the During actions have completed on the service component and any associated resource offerings. Actions configured in the After phase on a resource offering run after all of the After actions on the associated service component have completed.

- **On Failure** - Actions configured in the On Failure phase on a service component or resource offering run after an action failure occurs in any of the Before, During, or After phases for that service component or resource offering.

Tasks

- **Navigate to** (the Lifecycle tab) - In the left pane of the **Sequenced Components** area, select the palette associated with the component type of the component template you want to view. Select the component type and then select the **Templates** tab. Select the component template, and then select the **Lifecycle** tab.

- **Add Action** - Click the **Add Action** or **Add More Actions** link in the stage and phase to which you want to add an action. Enter the information as described in "Lifecycle Action Selection Wizard" on the next page.

- **Edit Action** - Select the gear icon on the action you want to edit and select **Edit**. Update the information as described in "Lifecycle Action Selection Wizard" on the next page.

- **Delete Action**- Click the gear icon next to the action you want to delete and select **Delete**. You can also delete an action that is in a group.

- **Move Action**- Select the gear icon on the action or group of actions you want to move to another stage or phase and select **Move**. In the Move dialog, select the Target Stage and Target Phase from the pull-down menus and then click **Move**. You can also click and drag to move an action or group of actions. To move an action or group of actions using the keyboard, press the tab key to provide focus on the action or group and then press Ctrl and the up or down keys to move the action or group within a phase.

- **Group Actions** - Click to select multiple actions in a lifecycle phase and then click **Group Actions** to create a parallel execution group. Actions in a parallel execution group are executed at the same time.

- **Ungroup Actions** - Click the gear icon next to an action in a group and then select **Remove from Group**.

- **View Lifecycle Stages** - The Deploying Provisioning stage, the Modifying Operational stage the Un-deploying De-provisioning stage, and any stage that has an action, are displayed by default. Select a filter from the All Selected Stages pull-down menu to display only Provisioning Stages, Operational Stages, or De-provisioning stages. Click **Select Stages** to view additional stages.

- **View Lifecycle Phases** - The During and On Failure phases are displayed by default. Select **Show All Phases** to display the Before and After phases.

Lifecycle Action Selection Wizard

See also Lifecycle Action Selection Wizard (for service components) or Lifecycle Action Selection Wizard (for resource offerings).

Concepts

Use the action selection wizard to create or edit an action for the selected lifecycle phase.

To add Operations Orchestration flows, see the Import Operations Orchestration Flows section in the *Configuration Guide* for more information.

Tasks

- **Select Action** - Search for flows or actions by name (when searching for a flow, the folders searched in the Operations Orchestration library are determined by a property configured in the `csa.properties` file; see the "Action Selection Wizard" property description section in the Configuration Guide for more information) or select the process engine from which to select a flow or action. Then, locate and select the flow or action. See "Lifecycle Action Internal Actions" on page 212 for a list of internal actions. This task is only available when you create an action.

- **Configure Action** - Configure information about the action.

Item	Description
Process Engine	Read-only. The container of the flow or internal action.
Locator	Read-only. The location of the flow or internal action.
Display Name	The name you provide for the action.
Description	The description you provide for the action.
Execution Properties	Specify the following: ○ Fail on Error - If selected, provisioning or de-provisioning will stop if the action fails. By default, Fail on Error is not selected for actions created in the Un-deploying, Un-reserving, and Un-initializing stages. ○ Error on Timeout - If selected, provisioning or de-provisioning will stop if the time

Item	Description
	to execute the action exceeds the timeout value, configured below.
Timeout (seconds)	The amount of time to wait for the action to complete. Set this field to zero (0) if you do not want the action to time out. Note that this creates the potential of an action not finishing, which may cause the underlying subscription to remain in the Pending state.

- **Configure Parameters** - Configure the input parameters whose values are passed to the flow or internal action prior to invoking the action. Click the gear icon next to the parameter you want to configure and select **Edit Mapping**. From the Edit Parameter Mapping dialog, edit the Name, Display Name, Description, and Mapping Type of the parameter.

  Select one of the following Mapping Types:

  - Not Mapped - The parameter is not mapped to a value.

  - Constant Value - Type a value for the parameter.

Lifecycle Action Internal Actions

For more information about lifecycles, see "Lifecycle Actions for Component Templates" on page 207.

The following internal actions ship with Codar:

Internal Action	Description	Applies To
Build Resource Provider and Pool List	Builds a candidate list of resource providers and associated resource pools that meet the following requirements: • Support the resource offering. • Have an **Availability** of **Enabled**. • The provider's resource pool has sufficient resource capacity. To determine this, you must consider all measurable properties as configured in the **Measurable Properties** tab for the resource offerings, as well as the optional **Multiplier Property Name** field. The resource pool must have enough resource capacity to support all the properties, which requires that each necessary resource type (for example: CPU, Memory, and Storage), based on the measurable properties, be configured on the pool either with a **Resource Availability** of **Unlimited** or **Available**. If **Available**, the difference between **Total Available To CSA** and **Current CSA Utilization** must be sufficient to support the measurable property requirements.	Resource Offering associated with a Service Component
Build Resource Provider List	Builds a candidate list of resource providers that meet the following requirements: • Support the resource offering. • Have an **Availability** of **Enabled**.	Resource Offering associated with a Service Component
Clone Pattern	Clones a service component that is marked as a **Pattern** into one or more non-pattern service components. The number of service components created is determined by the value of the property specified in **Name of the Property for Service Component Count**.	Service Component
Decrease Resource Utilization	Decreases the utilization of one or more resources in a resource pool by the values of the measurable properties configured on a resource offering associated with a service component. **Note:** The Decrease Resource Utilization action cannot be added to a component, component template, or resource offering.This action runs automatically during Un-reserving when Resource Accounting is enabled for a resource offering associated with a service component.	Resource Offering associated with a Service Component
Increase	Increases the utilization of one or more resources in a resource pool by	Resource

Internal Action	Description	Applies To
Resource Utilization	the values of the measurable properties configured on a resource offering associated with a service component. **Note:** The Increase Resource Utilization action cannot be added to a component, component template, or resource offering. This action runs automatically during Reserving when Resource Accounting is enabled for a resource offering associated with a service component.	Offering associated with a Service Component
Log Messages	Writes the user-specified Boolean Input, Integer Input, and String Input property values to the `csa.log` file. You can include this as an action on a Service Component, Resource Offering associated with a service component, or Resource Offering for use in troubleshooting.	Resource Offering Resource Offering associated with a Service Component Service Component
Select Resource Provider	Selects a resource provider from the candidate list that was built by the **Build Resource Provider List** action. The selected resource provider will be available to resource offering actions in the token `RSC_PROVIDER_ID`. The selected provider will, optionally, be written to a property on the associated service component if the `Provider Property Name` input to the action is provided.	Resource Offering associated with a Service Component
Select Resource Provider and Pool	Selects a resource pool and provider from the candidate list that was built by the **Build Resource Provider and Pool List** action. The selected resource provider and pool will be available to resource offering actions in the token `RSC_PROVIDER_ID` and `RSC_POOL_ID`, respectively. The selected pool will, optionally, be written to a property on the associated service component if the `Pool Property Name` input to the action is provided.	Resource Offering associated with a Service Component
Select Resource Provider and Pool from Parent	Selects the resource pool and provider already chosen by a service component's parent service component, as identified by the `Parent Component ID` and `Pool Property Name` properties. The selected resource provider and pool will be available to resource offering actions in the token `RSC_PROVIDER_ID` and `RSC_POOL_ID` respectively. The selected pool will also be written to a `Pool Property Name` property on the associated service component.	Resource Offering associated with a Service Component
Select Resource Provider from	Selects the resource provider already chosen by a service component's parent service component, as identified by the `Parent Component ID` and `Provider Property Name` properties. The selected resource provider will be available to resource offering actions in the token `RSC_`	Resource Offering associated with a

Internal Action	Description	Applies To
Parent	`PROVIDER_ID`. The selected resource provider will also be written to a `Provider Property Name` property on the associated service component.	Service Component

User Operations for Component Templates

User operations are actions configured on resource offerings or service, component templates, components in a design that can be invoked by users in the Operations area after a design has been provisioned. For example, a Restart Server user operation can be configured on a server service component or on an associated resource offering, allowing a user to restart a particular provisioned server after the design has been provisioned. User operations can also be invoked on a service instance in the Operations area.

Tasks

- **Navigate to** (the **User Operations** tab) - In the left pane of the **Sequenced Components** area, select the palette associated with the component type of the component template you want to view. Select the component type and then select the **Templates** tab. Select the template, and then select the **User Operations** tab.

- View a list of user operations for this component template.

- **Add Action** - Click **Add**, and enter the information as described in "User Operation Action Selection Wizard" on the next page.

- **Edit Action** - Select the gear icon on the action you want to edit, and select **Edit**. Update the information as described in "User Operation Action Selection Wizard" on the next page.

- **View User Operation** - Click the **User Operations** tab to view all user operations associated with a component template.

- **Delete User Operation** - Click the gear icon next to the user operation you want to remove, and select **Delete**.

User Operation Action Selection Wizard

Use the action selection wizard to create or edit a user operation.

To add Operations Orchestration flows, see the Import Operations Orchestration Flows section in the *Configuration Guide* for more information.

Tasks

- **Select Action** - Search for flows or actions by name (when searching for a flow, the folders searched in the Operations Orchestration library are determined by a property configured in the `csa.properties` file; see the "Action Selection Wizard" property description section in the Configuration Guide for more information) or select the process engine from which to select a flow or action. Then, locate and select the flow or action.

- **Configure Action** - Configure information about the action.

Item	Description
Process Engine	Read-only. The container of the flow or internal action.
Locator	Read-only. The location of the flow or internal action.
Display Name	The name you provide for the action.
Description	The description you provide for the action.
Execution Properties	Specify the following: ○ Fail on Error - During provisioning or de-provisioning, the Fail On Error setting indicates whether provisioning or de-provisioning should continue after an action has failed. For a user operation, the Fail on Error setting is irrelevant because there are no subsequent actions that would run after the user operation. ○ Error on Timeout - If selected, the user operation will be considered to have failed if the time to execute the action exceeds the Timeout value (see below).
Timeout (seconds)	The amount of time to wait for the action to complete. Set this field to zero (0) if you do not want the action to time out.

- **Configure Parameters** - Configure the input parameters whose values are passed to the flow or internal action prior to invoking the action. Click the gear icon next to the parameter you want to configure and select **Edit Mapping**. From the Edit Parameter Mapping dialog, edit the Name, Display Name, Description, and Mapping Type of the parameter.

Select one of the following Mapping Types:

- Not Mapped - The parameter is not mapped to a value, and no value will be provided for this parameter when the action is run.

- Constant Value - Type a value for the parameter. Note that a user will be unable to edit this value.

- Prompt User - Allows the user to enter a value for the parameter when running the user operation. A default value for the parameter is required.

- Prompt User List - Allows the user to select a value for a parameter by selecting from a list of items when running the user operation.

List Items	
Switch to Static Entry (default)	Enter the display name and value for a list item. After entering a name or value, input boxes for a new list item automatically display.
Switch to Dynamic Entry	- Select a script from the List Items. **HTTP Parameters** - Specify the HTTP parameters to use when executing the script. The required input parameters are dependent on the script that has been selected; contact the script developer if necessary to determine the necessary input parameters. **Refresh Data** - Click the Refresh Data icon ( ) to run the script and display the results.

Associate Resource Offerings with Component Templates

You can associate a resource offering with a component template. This association or link ensures that the resource offering will be provisioned when the Server service component is deployed.

Tasks

- **Navigate to** (the Resource Offerings tab) - In the left pane of the **Sequenced Components** area, select the component palette that contains the component type whose template resource offerings you want to view. Click the component type. Select the **Templates** tab. From the templates list, click the component template whose resource offerings you want to view. Click the **Resource Offerings** tab to review the list of resource offerings.

- **Add** - From the Resource Offerings tab, click the **Add** icon. In the **Select Resource Offering** dialog, select the **Resource Category** from the list, select **Resource Offering** from the list, and click **Add**.

  A resource offering cannot be associated with a service component if the service component is part of a published design.

- **Navigate to Resource Offering** – Click the gear icon next to the resource offering to which you want to navigate. Select **Navigate to Resource Offering**, which takes you to the resource offering.

- **Replace Resource Offering** - Click the gear icon next to the resource offering you want to replace. Select **Replace Resource Offering**, and select the replacement resource offering. The displayed resource offerings are not bound to the service component and are of the same resource category and provider type as the resource offering being replaced.

- **Delete** - Click the gear icon next to the resource offering you want to delete. Click **Delete** and click **Yes** in the confirmation dialog.

  A resource offering cannot be deleted if the design is in an upgrade path, is upgradable from a previous version of the design, and the resource offering existed in that prior version. Resource offerings that are newly added to the current version of the design can be deleted.

  A resource offering cannot be deleted if the service component is part of a published design.

- **Group** – Click to select multiple resource offerings, and click **Group** on the right panel to create a parallel processing group. You can also click and drag individual resource offerings to group them. Resource offerings in a parallel processing group are executed at the same time.

- **Ungroup** – Click the gear icon next to a resource offering inside a group and select **Remove from Group**. You can also ungroup a resource offering by dragging it out of the parallel processing group.

  - To ungroup multiple resource offerings at one time, click and select multiple resource offerings and click **Ungroup** on the right panel.

- **Set Processing Order** – The processing order of a resource offering specifies the order in which the associated resource offering will be provisioned relative to the other resource offerings added on this service component. The order in which you see the resource offerings are the order in which they will be provisioned. You can click and drag a resource offering or a group of resource offerings and move them around to set processing order.

Provider Selection Panel

For more information about associating resource offerings to service component templates, see
"Associate Resource Offerings with Component Templates" on page 218.

Concepts

After adding a resource offering to a component template, you must create one or more provider
selection actions for that resource offering. Be sure to do this if you have configured a resource
environment in order to control which resource providers should be used when service designs are
deployed. Otherwise, the default, random selection of a resource provider that occurs when provider
selection actions are not called will ignore the resource environment association. The provider selection
actions execute during the Before phase of the Reserving lifecycle stage; the lifecycle stage cannot be
changed.

Codar ships with internal actions to help with provider selection. For more information about these
provider selection internal actions, see "Provider Selection Internal Actions" on page 222.

Tasks

- **Navigate to** or **View** (the Provider Selection panel) - From the **Sequenced Designs** page, select a
  service design, and then select the design version. Click the **Designer** tab. Select the service
  component on which you want to add a provider selection action. In the right pane, click the gear
  icon, select **Edit Component**, and select the **Resource Offerings** tab. In the **Resource Offerings**
  tab, click the name of the resource offering, and then click **Provider Selection** from the panel on
  the right.

- **Add** (a provider selection action) - Navigate to the **Provider Selection** panel and click **Add**. If no
  provider selection actions are configured, click the **Add** icon in the middle of the panel. Enter the
  information as described in the "Lifecycle Action Selection Wizard" on page 210.
  A provider selection action cannot be created if the service component is part of a published design.

- **Edit** (a provider selection action) - Navigate to the **Provider Selection** panel, click the gear icon
  next to the action you want to edit, and select **Edit**. Update the information as described in the
  "Lifecycle Action Selection Wizard" on page 210.
  The properties of an action cannot be edited if the service component is part of a published design.

- **Delete** (a provider selection action) - Navigate to the **Provider Selection** panel, click the gear icon
  next to the action you want to delete, and select **Delete**.

- **Set Processing Order** – The processing order of a provider selection action is relative to the other provider selection actions on this resource offering. The order in which you see the provider selection actions is the order in which they will be executed. You can click and drag a provider selection action to change its processing order.

Provider Selection Internal Actions

For more information about provider selection actions, see "Provider Selection Panel" on page 107.

Concepts

CSA ships with internal actions to help with provider selection.

- If you have *not* configured resource pools on a provider, then the two provider selection actions most likely to be configured on a resource offering are the **Build Resource Provider List** action and the **Select Resource Provider** action, which should run in that order. To select the provider already selected by the parent service component, use the **Select Resource Provider from Parent** action instead of the two previously mentioned actions.

- If you have configured resource pools on a provider, then the two provider selection actions most likely to be configured on a resource offering are the **Build Resource Provider and Pool List** action and the **Select Resource Provider and Pool** action, which should run in that order. To select the provider and pool already selected by the parent service component, use the **Select Resource Provider and Pool from Parent** action instead of the two previously mentioned actions.

See the following table for descriptions of the out-of-the-box internal actions that ship with Codar:

Internal Action	Description	Applies To
Build Resource Provider and Pool List	Builds a candidate list of resource providers and associated resource pools that meet the following requirements: - Support the resource offering. - Have an **Availability** of **Enabled**. - The provider's resource pool has sufficient resource capacity. To determine this, you must consider all measurable properties as configured in the **Measurable Properties** tab for the resource offerings, as well as the optional **Multiplier Property Name** field. The resource pool must have enough resource capacity to support all the properties, which requires that each necessary resource type (for example: CPU, Memory, and Storage), based on the measurable properties, be configured on the pool either with a **Resource Availability** of **Unlimited** or **Available**. If **Available**, the difference between **Total Available To CSA** and **Current CSA Utilization** must be sufficient to support the measurable property requirements.	Resource Offering associated with a Service Component

Internal Action	Description	Applies To
Build Resource Provider List	Builds a candidate list of resource providers that meet the following requirements: • Support the resource offering. • Have an **Availability** of **Enabled**.	Resource Offering associated with a Service Component
Clone Pattern	Clones a service component that is marked as a **Pattern** into one or more non-pattern service components. The number of service components created is determined by the value of the property specified in **Name of the Property for Service Component Count**.	Service Component
Decrease Resource Utilization	Decreases the utilization of one or more resources in a resource pool by the values of the measurable properties configured on a resource offering associated with a service component. **Note:** The Decrease Resource Utilization action cannot be added to a component, component template, or resource offering.This action runs automatically during Un-reserving when Resource Accounting is enabled for a resource offering associated with a service component.	Resource Offering associated with a Service Component
Increase Resource Utilization	Increases the utilization of one or more resources in a resource pool by the values of the measurable properties configured on a resource offering associated with a service component. **Note:** The Increase Resource Utilization action cannot be added to a component, component template, or resource offering. This action runs automatically during Reserving when Resource Accounting is enabled for a resource offering associated with a service component.	Resource Offering associated with a Service Component
Log Messages	Writes the user-specified Boolean Input, Integer Input, and String Input property values to the `csa.log` file. You can include this as an action on a Service Component, Resource Offering associated with a service component, or Resource Offering for use in troubleshooting.	Resource Offering Resource Offering associated with a Service Component Service Component
Select Resource Provider	Selects a resource provider from the candidate list that was built by the **Build Resource Provider List** action. The selected resource provider will be available to resource offering actions in the token `RSC_PROVIDER_ID`. The selected provider will, optionally, be written to a property on the	Resource Offering associated with a

Internal Action	Description	Applies To
	associated service component if the `Provider Property Name` input to the action is provided.	Service Component
Select Resource Provider and Pool	Selects a resource pool and provider from the candidate list that was built by the **Build Resource Provider and Pool List** action. The selected resource provider and pool will be available to resource offering actions in the token `RSC_PROVIDER_ID` and `RSC_POOL_ID`, respectively. The selected pool will, optionally, be written to a property on the associated service component if the `Pool Property Name` input to the action is provided.	Resource Offering associated with a Service Component
Select Resource Provider and Pool from Parent	Selects the resource pool and provider already chosen by a service component's parent service component, as identified by the `Parent Component ID` and `Pool Property Name` properties. The selected resource provider and pool will be available to resource offering actions in the token `RSC_PROVIDER_ID` and `RSC_POOL_ID` respectively. The selected pool will also be written to a `Pool Property Name` property on the associated service component.	Resource Offering associated with a Service Component
Select Resource Provider from Parent	Selects the resource provider already chosen by a service component's parent service component, as identified by the `Parent Component ID` and `Provider Property Name` properties. The selected resource provider will be available to resource offering actions in the token `RSC_PROVIDER_ID`. The selected resource provider will also be written to a `Provider Property Name` property on the associated service component.	Resource Offering associated with a Service Component

Resource Accounting Tab

For more information about associating resource offerings to service components, see "Associate Resource Offerings with Component Templates" on page 218.

Concepts

Resource accounting actions track the utilization of resources in a resource pool.

The following out-of-the-box accounting actions are added to the resource offerings when you enable resource accounting:

- **Increase Resource Utilization** - This action is configured to run during the **After** phase of the **Reserving** lifecycle stage.

- **Decrease Resource Utilization** - This action is configured to run during the **After** phase of the **Un-reserving** lifecycle stage.

Tasks

- **Navigate to** or **View** - In the left pane of the **All Designs** area, select the tag associated with the design you want to view, and select the service design. In the **Designer** tab, select the service component whose resource accounting action properties you want to view. In the right pane, click the gear icon and select **Edit Component**. In the **Resource Offerings** tab, select the resource offering. The panel on the right will display information about **Resource Accounting**.

- **Enable** (Resource Accounting) – Navigate to the **Resource Offerings** tab. Select the resource offering on which you want to enable resource accounting. Select the **Enable Resource Accounting** checkbox on the panel on the right. When you enable resource accounting, the two internal actions **Increase Resource Utilization** and **Decrease Resource Utilization** are automatically added to the resource offering.

- **Disable** (Resource Accounting) – Navigate to the **Resource Offerings** tab. Select the resource offering on which you want to disable resource accounting. Deselect the **Enable Resource Accounting** checkbox on the panel on the right. When you disable resource accounting, the two internal actions **Increase Resource Utilization** and **Decrease Resource Utilization** are removed from the resource offering.
Resource accounting cannot be enabled/disabled if the service component is part of a published design.

Best Practices

- Do not create resource accounting actions on group-level service components. For example, do not create a resource accounting action on a Server Group; instead, create it on a Server service component.

Measurable Properties Tab

For more information about associating resource offerings to service components, see "Associate Resource Offerings with Component Templates" on page 218.

Concepts

A measurable property is an integer service component property that has a configured **Resource Type** and **Unit** (see "Service Component Properties" on page 123). When you configure measurable properties on a resource offering, you create references to the corresponding service component measurable properties. To view the value of a measurable property, see the corresponding service component property in the **Properties** tab.

Measurable properties are used by the following out-of-the-box actions to assist in provider and pool selection and in resource accounting:

- Build Resource Provider and Pool List

- Increase Resource Utilization

- Decrease Resource Utilization

If you configure any of these actions on a resource offering in the **Provider Selection** or enabled **Resource Accounting**, you must also configure the measurable properties for that resource offering in the **Measurable Properties** panel.

Tasks

- **Navigate to** or **View** (the Measurable Properties panel) - In the left pane of the **All Designs** area, select the tag associated with the design you want to view. Select the service design and, in the **Designer** tab, select the service component that contains the resource offering whose measurable properties you want to view. In the right pane, click the gear icon and select **Edit Component**. In the **Resource Offerings** tab, click the name of the resource offering whose measurable properties you want to view. Finally, select the **Measurable Properties** panel on the right.

- **Add** - Navigate to the **Measurable Properties** panel and select the measurable properties you want to add. If the service component does not contain any measurable properties, the measurable properties panel will be empty.

A measurable property of a resource offering in a service component cannot be added if it is part of an upgraded or published design.

- **Delete** - Navigate to the Measurable Properties panel and de-select the measurable properties you want to remove.

A measurable property of a resource offering in a service component cannot be deleted if it is part of a published design or a design that is in an upgrade path, is upgradable from a previous version of the design, and the measurable property existed in that prior version.

Palettes

A palette is a group of related components that can be used in a topology design.

Tasks

After a palette is created, you can do the following from the Overview tab:

- **Save a palette.**

- **Edit a palette.** You can edit global properties, such as name and description.

- **Delete a palette.**

From the Components tab, you can click a component name to open it for editing. You can also:

- **Remove selected components from the palette.**

- **Add a component to the palette.** You can filter the list of available components by tag, palette, or provider type.

Automate Release

The Release Automation sidebar menu item defines the configuration tasks required to move a design across lifecycle stages. This sidebar menu item is further divided into four sidebar menu items:

- Release Pipeline - used to manage packages, create release gate actions, manage environments, and view pipeline statistics

- Pipeline Configurations - used to create and configure lifecycle stages, create roles, associate roles with a fixed set of permissions, and associate lifecycle stages with the roles.

- Approvals - used to manage approval requests for approval actions. The approver can view the approval requests for stage transition and approve or reject these requests.

- Server configurations and pool creations - used to configure existing servers for application deployment and creation of pools to be associated with lifecycle stages.

Refer to the following topics for more information:

- "Packages" on the next page

- "Release pipeline" on page 248

- "Pipeline Configurations" on page 308

- "View different versions for a single application design" on page 297

- "Approve or deny an approval request" on page 315

- "Configure servers" on page 316

- "Associate pools" on page 321

- "Associate server pools with lifecycle stages" on page 327

Packages

Packages represent a snapshot of an application design and allow properties to be parameterized within the design. We can also say that the package represents a particular build of an application.

A package is the smallest unit that can be deployed for an application. It represents both the implementation artifacts (the manner in which an application should be deployed) and deployment artifacts(the location of libraries like war, ear, etc., that should be deployed).

Packages are associated with a lifecycle stage. A package can belong to Development, Testing, Staging, or Production stages.

Packages are associated with pipeline management. They can be managed across lifecycle stages, such as promotion or rejection in a given stage. For example, a user with the QA role can reject a package. See "Lifecycle stages and actions" on page 309.

Tasks

- **Create a package from a specific application version.** An application version can consist of multiple packages. See "Create a package" on page 233.

- **Deploy or redeploy a package.** In this case the corresponding state of an application design along with the properties of the design specified in the package will be fulfilled. See "Deploy a package" on page 236.

- **Delete a package.** Go to the Release Pipeline tab, hold Ctrl to select multiple packages, and click **Delete**.

  You cannot delete a package that has an instance associated with it.

View packages

For more information about packages, see "Packages" on the previous page.

To view packages

1. Click the **Release Pipeline** sidebar menu item.

2. Select the type of application design in which you need to view a package, from the drop-down list on the left side of the screen. (**Sequenced Designs** is selected by default)

3. Click the application design version.

    All packages pertaining to their respective lifecycle stage for the selected application design version are displayed.

4. Click the package you want to view.

5. You can use the search box on the right to find a specific package. Filter packages by stage and state using the drop-down lists on the right.

Packages are grouped into stages, with three packages displayed per stage initially. Click **More** to the right of the stage heading to view all packages for a certain stage. For more information about stages, see "Lifecycle stages and actions" on page 309.

Create a package

For more information about packages, see "Packages" on page 231.

> You may not have access to some or all of this functionality, depending on your role. See Roles.

To create a package

1. Click the **Release Pipeline** sidebar menu item.

2. Select the type of application design (sequenced or topology) in which you need to create a package, from the drop-down list on the left side of the screen. (**Sequenced Designs** is selected by default).

3. Click the application design version for which you need to create the package.

4. Click the gear icon next to the drop-down list on the right side of the screen and select **Create Package**.

5. Enter a name and description for the package. The package name is displayed automatically. You can change it, if required.

   The application design and version are displayed, but cannot be changed.

6. Click **Create**. A package will be created in the Development stage.

> **Note:** List properties are available in packages during Release Automation, only for Sequenced Designs.

For more information about the package lifecycle, see Lifecycle stages and actions.

Edit a package

For more information about packages, see "Packages" on page 231"Packages" on page 231

You can edit a package name and description. For information about changing a package state or stage, see "Lifecycle stages and actions" on page 309.

> You may not have access to some or all of this functionality, depending on your role. See Roles.

To edit a package

1. Click the **Release Pipeline** sidebar menu item.

2. Select the type of application design in which you need to edit a package, from the drop-down list on the left side of the screen. (**Sequenced Designs** is selected by default).

3. Click the application design version. All packages pertaining to their respective lifecycle stage for the selected application design version are displayed.

4. Click the package you want to edit.

5. If you don't see the package you want to edit, see "Packages" on page 231.

6. Click the gear icon on the right and select **Edit**.

7. Change the package name or description.

8. Click **Save**.

Configure package properties

From Configure tab for a package, you can change the properties that can be modified for all components used in the design.

To configure package properties

1. Click the **Release Pipeline** sidebar menu item.

2. Select the type of application design in which you need to configure a package, from the drop-down list on the left side of the screen. (**Sequenced Designs** is selected by default).

3. Click the application design version. All packages pertaining to their respective lifecycle stage for the selected application design version are displayed.

4. Click the package you want to configure.

5. Select the **Configure** tab.

6. Click **Show Properties** on the right-hand side to view the properties.

7. Change the desired properties.

   Note: : Codar does not support list type of property for packages.

8. Click the gear icon on the right-hand side and select **Save**.

Note: The package properties configured in the above manner are applicable to the selected package only. When a new package is created, it will by default have the properties that are defined for the application design.

Deploy a package

For more information about packages, see "Packages" on page 231.

> You may not have access to some or all of this functionality, depending on your role. See Roles.

To deploy a package

1. Click the **Release Pipeline** sidebar menu item.

2. Select the type of application design in which you need to deploy a package, from the drop-down list on the left side of the screen. (**Sequenced Designs** is selected by default).

3. Click the application design version. All packages pertaining to their respective lifecycle stage for the selected application design version are displayed.

4. Click the gear icon pertaining to the package you want to deploy and select **Deploy**.

5. Enter a name for the deployment.

6. Select an environment for the deployment. For Sequenced Designs, you can select multiple environments.

7. (Only for partial designs). Click **Next** and select a service design from the list of available designs.

   Select a design from the list that, together with the partial design, will form a composite design that can be provisioned. If you select **Custom Selection**, you can select a microservice for each requirement in the partial design. Only service designs or microservices that fulfill the requirements are available.

8. (Optional) Click **Next** and change/specify modifiable properties for this deployment.

   For topology designs, default values of modifiable properties will be displayed automatically. You can change these values, if required.
   For sequenced designs, all properties with default value are displayed automatically.

9. Click **Deploy**.

When a package is deployed successfully in the final lifecycle stage, all its properties will be copied back to the design and will be saved automatically.

Deployments tab

After a package has been deployed, an entry is made on the **Deployments** tab for the package. This tab lists all deployments, their respective lifecycle stage, date and time of deployments, deployment type, their status, and the user who created the deployment. Click a deployment for quick access to the test results for a deployment.

Redeploy a package

You may want to redeploy a package in order to test a deployment with different properties, such as a different location or different credentials. You can redeploy a package instead of recreating the design. Package redeployment is only applicable to topology design.

For more information about packages, see "Packages" on page 231.

> You may not have access to some or all of this functionality, depending on your role. See Roles.

To redeploy a package

1. Click the Release Pipeline sidebar menu item.

2. Select the desired topology design.

3. Select a package.

4. Optionally create a new package.

5. Click the gear icon pertaining to the package and select **Redeploy** .

6. Enter a name for the deployment.

7. Select an existing deployment that you want to redeploy.

   After you select a deployment, its details will be displayed so you can verify that you selected the desired deployment.

8. Click **Next** and change modifiable properties as needed for the redeployment.

9. Click **Deploy**.

Deployments tab

After a package has been deployed, an entry is made on the Deployment tab for the package. This tab lists all deployments, their status, and the user who created the deployment. Click a deployment for quick access to the test results for a deployment.

Scaling a deployment

You can create multiple instances of a group for a deployment. Each component in the group will be duplicated by the number of instances you select. Scaling is only applicable to topology design.

To scale out

1. Click the Release Pipeline sidebar menu item.

2. Select the desired topology design.

3. Open the package you need to scale out.

4. Go to the Deployments tab.

5. Click **Scale Out**.

   This button is only available for successful deployments.

6. Select an instance size.

   This is the number of instances you want of each component in the scalable group.

7. Click **Scale Out** in the dialog to complete the operation.

Note: For infrastructure resource pools, the pools should have enough number of servers associated with them for the number of instances specified at the time of scale out.

Schedule package promotion from a lifecycle stage to the next

The **Schedules** tab lists the details of all automatic lifecycle stage promotions scheduled for a package.

From this tab, you can manage the schedules to automatically promote packages from a specific lifecycle stage to the next.

This tab displays the following details for a schedule:

- **Promotion Stage**: The stage from which the package was scheduled to be promoted.

- **Schedule Name**: Name of the package promotion schedule.

- **Schedule Date**: The date and time at which the promotion was scheduled to take place.

- **Scheduled By**: The user who created the schedule.

- **Description**: Schedule description.

- **Status**: Status of the schedule (pending/success/failure).

See "Manage Schedules" on the next page

Manage Schedules

The **Manage Schedules** screen lists the details of all automatic lifecycle stage promotions scheduled for a package.

You can perform the following tasks.

- "View existing schedules" below

- "Create new promote schedules" on page 243

- "Edit existing promote schedules" on page 244

- "Delete existing promote schedules" on page 245

View existing schedules

1. Go to **Release Automation** > **Release Pipeline**.

2. Select the design type (sequenced or topology) from the drop-down list in the left pane.

3. Select a design and click the design version.

4. In the **Release Pipeline** tab, you will find the list of packages in various stages. Click on the name of the package.

5. Click **Schedules** tab.

6. Click **Manage Schedules** button. You can find the list of schedules maintained for the package in the **Manage Schedules** screen.

View lifecycle stage promotion schedules

You can view a list of all lifecycle stage promotions scheduled for packages of a particular application version, from the **Schedules** tab. You can view month or day-wise package promotion schedules.

To view package promote schedules

1. Click the **Release Pipeline** sidebar menu item.

2. Select the type of application design in which you need to view the promotion schedule, from the drop-down list on the left hand side (**Sequenced Designs** is selected by default).

3. Click the version of the application design which contains packages scheduled for promotion.

4. Click the **Schedules** tab.

   By default, the application displays a calendar view of month-wise schedules, showing dates on which a promote has been scheduled along with schedule name and time of schedule, for the month. You can switch between month and day views.

Create new promote schedules

To create a new package promote schedule

1. In the **Schedules** tab, click **Manage Schedules** button to open **Manage Schedules** screen.

2. Click the 'add schedule' button.

3. **Add Schedule** window pops up. Specify the following details in the **Add Schedule** window.
   a. Display name for the schedule.

   b. A brief description of the schedule.

   c. Schedule date and time in **MM/DD/YYYY HH:MM AM/PM** format. Depending on the time zone (IST, PST, etc), the system displays the current time on the server on which Codar has been hosted.

   d. Lifecycle stage from which the package should be promoted on the above schedule.

   e. | **Note:** Schedule date cannot be a date in the past.

      Promotion schedule date for a lifecycle stage cannot be earlier than the schedule date set for a previous lifecycle stage.

4. Click Add button.

5. Click 'Done' to exit the **Manage Schedules** screen.

Edit existing promote schedules

To edit an existing package promote schedule

1. In the **Schedules** tab, click **Manage Schedules** button to open **Manage Schedules** screen.

2. Select the schedule you want to edit.

3. Click 'edit scheduling' button.

4. Change the details of the schedule.

5. Click **Save** button.

6. Click 'Done' to exit the **Manage Schedules** screen.

Delete existing promote schedules

Only users with 'Admin' role or the schedule creator can delete promote schedules.

To delete an existing package promote schedule

1. In the **Schedules** tab, click **Manage Schedules** button to open **Manage Schedules** screen.

2. Select the schedule you want to delete.

3. Click 'delete schedule' button.

4. Click 'Yes' to confirm deletion.

5. Click 'Done' to exit the **Manage Schedules** screen.

Release Gates tab

The **Release Gates** tab, by default, shows the following information of the action instance(s) related to the latest release gate of the package.

Field	Details
Action Name	Name of the action
Action Type	Type of action (deploy, custom, approval or test set)
Start Time	Date and time when the action started
End Time	Date and time when the action ended
Status Message	Detailed message related to the status of the action
Status	Status of the action

Under **Release Gates Action Instances**, the user can click the link pertaining to an action name to view details associated with it for deploy, test set, and OO flow actions. For approval actions, clicking the link will open **Approvals** page for further action by the user.

For OO flow actions, clicking the respective action under 'Action Name' opens the OO flow report. Here, you can see the status of the action, step name, transition message, OO flow graph, and step details (input and output details).

For 'Provision Cloud Service' actions, clicking the respective action under 'Action Name' opens a separate login page, where you must log in to view realized subscription. This happens since both the CSA and Codar instances are not sharing the same IDM. You just need to log in once, and it will be valid till your current session lasts. If both CSA and Codar instances are sharing the same IDM, you will not be required to log in separately.

Click the **Release Gates** link to view how the package moved from one lifecycle stage to another along with date, time, and status of each movement.

Here, you can pause or stop execution of the release gates which are in 'In Progress' status, or delete the release gates which have been stopped, by selecting the relevant option from the gear icon on the right hand side.

Note: You can pause or stop a release gate only when at least one of its action instances is in **Initiated** status.

In the **Release Gates** page, click the link pertaining to any of the listed release gates to view details of action instances associated with it.

View action instance properties

You can view all the realized values of a release gate action instance. To do this:

1. Navigate to **Release Automation** > **Release Pipeline**.

2. Select the design type (sequenced or topology) from the drop-down list in the left pane. (**Sequenced Designs** is selected by default).

3. Select the required application and version.

4. Select the package on which a release gate action has been executed.

5. Click **Release Gates** tab.

6. Click the gear icon next to the status of the action instance in and select the **Show Properties** option.

The 'Action Instance Properties' dialog appears containing following details:

- Action Name
- Action Type
- Action Order
- Action Instance Id
- Output Properties

Release pipeline

This sidebar menu item displays information for Topology Designs and Sequenced Designs .

Topology designs

Here, only designs that have been marked as application or microservice are displayed.

Application designs

You can view the list of application designs on the left pane. The designs that you can see may be limited by access control that is configured for each design.

The **Versions** tab lists all versions of an application design and the number of packages in each lifecycle stage.

The **Pipeline Statistics** tab displays a graphical representation of package data and equips users to make informed decisions based on the statistics.

The **Access Control** tab shows all users and groups who have been granted access to the design. This is a read-only view and can only be changed in the Design area.

Microservices

When a new lifecycle stage is created, a new microservice option is added. To view microservices, select a

lifecycle stage microservice in the upper right area of the page.

Sequenced Design

Here, all available sequenced designs are displayed.

The **Versions** tab lists all versions of a sequenced design and the number of packages in each lifecycle

stage.

The **Pipeline Statistics** tab displays a graphical representation of package data and equips users to make

informed decisions based on the statistics.

The **Access Control** tab shows all users and groups who have been granted access to the design. This is a

read-only view and can only be changed in the Design area.

View different versions for a single application design

The **Release Automation** > **Release Pipeline** >**Topology Designs or Sequenced Design**> **Versions** tab displays different versions of a selected application or sequenced design. It also displays all the lifecycle stages present in the design and the number of packages present in each stage. You can drill-down further and view the package overview and workflow by clicking a version number.

View the package workflow

The Package Workflow screen displays all the lifecycle stages in an application or sequenced design and the state of the packages in each lifecycle stage. To view the Package Workflow screen:

1. Navigate to **Release Automation** > **Release Pipeline**.

2. Select the type of design (sequenced or topology) for which you need to view the package workflow, from the drop-down list on the left side of the screen. (**Sequenced Designs** is selected by default).

3. Click the design and version for which you want to view the package workflow.

   The Package Workflow screen is displayed. This screen has several views:

   ○ View by Packages: Select **View by Packages** from the drop-down list on the right side of the screen to view all the packages in the design. Packages are displayed in their respective lifecycle stages. You can further filter by package states by selecting a package state from the drop-down list next to the **View by Packages** list.

   ○ View by Gate Actions: Select **View by Gate Actions** to view Release Gate Actions configured in respective lifecycle stages. You can view the Release Gate Actions in their respective lifecycle stage. Click an action to go to the **Edit Action** window.

   ○ View by Environments: Select **View by Environments** to view the selected Resource Environments in respective lifecycle stages.

   ○ View by Infrastructure Resource Pools: Select **View by Infrastructure Resource Pools** to view different resource pools configured across different lifecycle stages using the 'Configure Pools' option.

4. You can configure or create "Release gate actions" on page 261 for each lifecycle stage.

5. You can associate an environment with each lifecycle stage.

6. You can configure lifecycle stages by adding or removing a lifecycle stage. The lifecycle stage you add or remove applies only to the specific version you select. To configure lifecycle stages, select the gear icon on the right side of the screen and click **Configure Lifecycle Stages**.

Package statistics

The following statistics are displayed for each package available in the Package Workflow page.

- **Deployments Success:** The number of times the package has been deployed successfully.

- **Deployments Failed:** The number of times, the package deployment failed.

- **Deployments In progress:** The number of package deployments in progress currently.

- **Active/Transitioning/Rejected:** Package status, any one status displayed at a given time.

Associate an environment with a lifecycle stage

You can select resource environments that already exist and associate it with lifecycle stages. This helps you choose a subset of deployment environments to be associated with the lifecycle stage. By default all resource environments are associated with all lifecycle stages.

> **Note**: This option is only applicable to topology designs.

To select an environment to associate

1. Navigate to **Release Automation** > **Release Pipeline**.

2. From the list of topology designs on the left hand side, select the application design and version in which you select the environment.

3. Select the gear icon in the lifecycle stage with which you want to associate the environment and click **Select Environments**.

4. The Select Environments for <stage> dialog box displays all available resource environments. Select those that you want to associate and click **Add**. Selected environments are displayed in the Selected Environments box.

5. You can even search for environments using the Search fields. Type the name of the environment that you want to search for and press the Enter key.

6. Click **Save** to add the selected environments.

Configure purge schedules for packages and deployments

You can configure a schedule to automatically purge or remove packages and deployments periodically. This helps in freeing up of resources being held by packages/deployments that are no longer needed.

Users with 'Application Architect' or 'Admin' roles only can create or modify purge schedules.

You can perform the following tasks:

- "To view existing purge schedules" below

- "Create new purge schedules" on page 301

- "Edit existing purge schedules" on page 302

- "Delete existing purge schedules" on page 303

To view existing purge schedules

1. Go to **Release Automation** > **Release Pipeline**.

2. Select the design type (sequenced or topology) from the drop-down list in the left pane. (**Sequenced Designs** is selected by default).

3. Select the application design and version to view its purge schedule.

4. Click the gear icon corresponding to the lifecycle stage in which you need to configure the purge schedule, and select **Configure Purge Schedule**. The **Configure Purge Schedule for <lifecycle stage>** screen opens, displaying the list of purge schedules defined.

Create new purge schedules

To create a new deployment/package purge schedule

> **Note:** : Users with 'Application Architect' or 'Admin' roles only can create purge schedules.

1. Go to **Release Automation** > **Release Pipeline**.

2. Select the design type (sequenced or topology) from the drop-down list in the left pane. (**Sequenced Designs** is selected by default).

3. Select the application design and version in which you want to create a purge schedule.

4. Click the gear icon corresponding to the lifecycle stage in which you want to create the purge schedule, and select **Configure Purge Schedule**.

5. In the **Configure Purge Schedule for <lifecycle stage>** screen, click **Add Scheduler** and select the type of purge schedule you want to configure (deployment purge or package purge).

6. Specify the following details:

Field	Description
Display Name	Name of the purge schedule.
Purge <Deployments/Packages> Older than (in days)	Number of days after which deployments/packages in the given lifecycle stage must be purged.
Notification before (in days)	Number of days before which a notification must be sent about the impending purge.
Start Purge From	The date and time from which the schedule becomes active, in MM/DD/YYYY HH:MM AM/PM format. **Note:** Purge schedule date cannot be a date in the past. For a lifecycle stage, only one purge package schedule and one purge deployment schedule each can be created.

7. Click **Add Scheduler**.

8. Click **Done** to exit the **Configure Purge Schedule for <lifecycle stage>** screen.

Edit existing purge schedules

To edit an existing purge schedule

> **Note:** : Users with 'Application Architect' or 'Admin' roles only can modify purge schedules.

1. Go to **Release Automation** > **Release Pipeline**.

2. Select the design type (sequenced or topology) from the drop-down list in the left pane. (**Sequenced Designs** is selected by default).

3. Select the application design and version which contains the purge schedule you want to edit.

4. Click the gear icon corresponding to the lifecycle stage which contains the purge schedule you want to edit, and select **Configure Purge Schedule**.

5. Click the gear icon corresponding to the purge schedule you want to edit and select **Edit**.

6. Make the required changes and click **Save**.

7. Click **Done** to exit the **Configure Purge Schedule for <lifecycle stage>** screen.

Delete existing purge schedules

Only users with 'Admin' role or the schedule creator can delete purge schedules.

To delete an existing purge schedule

1. Go to **Release Automation** > **Release Pipeline**.

2. Select the design type (sequenced or topology) from the drop-down list in the left pane. (**Sequenced Designs** is selected by default).

3. Select the application design and version that contains the purge schedule you want to delete.

4. Click the gear icon corresponding to the lifecycle stage that contains the purge schedule you want to delete, and select **Configure Purge Schedule**.

5. Click the gear icon corresponding to the purge schedule you want to delete and select **Delete**.

6. Click **Yes** to confirm deletion.

7. Click **Done** to exit the **Configure Purge Schedule for <lifecycle stage>** screen.

View and cancel deployments

You can view all the deployments that you triggered during any lifecycle stage and choose to remove any of those deployments that are no longer relevant.

To view and remove deployments

1. Navigate to **Release Automation** > **Release Pipeline**.

2. Select the desired application design type (sequenced or topology), from the drop-down list on the left side of the screen. (**Sequenced Designs** is selected by default).

3. Select the application design and the version for which you want to view the deployments.

4. Click the **Deployments** tab.

   All the deployments that you triggered are displayed in a tabular format along with details such as the packages in each deployment, the lifecycle stage in which the deployment was triggered, and the status of the deployment.

5. You can filter the deployments based on lifecycle stages to view deployments pertaining to a specific lifecycle stage.

6. To remove a deployment, click the gear icon and click **Cancel**.

   All the packages in the deployment are listed in the Cancel Deployments dialog box. You can choose to remove one package or multiple packages.

   When you try to cancel a deployment that has a custom action associated with a deploy action, a warning message is displayed stating that upon cancellation, the custom action may not work. You can proceed to cancel the deployment by clicking 'Yes' in the message dialog box.

   > To delete a deployment, you must first cancel it.

7. Click **Cancel Deployments** and then **Yes** in the Confirmation Required confirmation box.

8. Click the gear icon and click **Delete**.

9. Select the packages to delete and click **Delete Deployments** in the Delete Deployments dialog box.

> **Note:** You can cancel the package deployment of another user if you have the 'Cancel and Delete Deployments' permission, provided the deployment is being canceled in the same lifecycle stage in which it was created. Example: If a user, who has Development role associated with Development lifecycle stage, deploys a package, another user with 'Cancel and Delete

Deployments' permission can cancel that deployment in the Development lifecycle stage itself.

Configure lifecycle stages

You can configure lifecycle stages by adding or removing a lifecycle stage to the existing subset of lifecycle stages. The lifecycle stage you add or remove applies only to the specific version you select.

To configure lifecycle stages:

1. Go to **Release Automation** > **Release Pipeline**.

2. Select the design type (sequenced or topology) from the drop-down list in the left pane. (Sequenced Designs is selected by default).

3. Select the application design and version for which you want to configure the lifecycle stages.

4. Select the gear icon on the right side of the screen and click **Configure Lifecycle Stages**.

5. Click **Add Lifecycle Stage** to add a lifecycle stage to the existing set of lifecycle stages.

6. To remove a lifecycle stage, click the bin icon next to the lifecycle stage that you want to remove.

7. To change the order of the lifecycle stages, drag and drop them above or below each other in the order you want.

8. Click **Save**.

View lifecycle stage promotion schedules

You can view a list of all lifecycle stage promotions scheduled for packages of a particular application version, from the **Schedules** tab. You can view month or day-wise package promotion schedules.

To view package promote schedules

1. Click the **Release Pipeline** sidebar menu item.

2. Select the type of application design in which you need to view the promotion schedule, from the drop-down list on the left hand side (**Sequenced Designs** is selected by default).

3. Click the version of the application design which contains packages scheduled for promotion.

4. Click the **Schedules** tab.

   By default, the application displays a calendar view of month-wise schedules, showing dates on which a promote has been scheduled along with schedule name and time of schedule, for the month. You can switch between month and day views.

Release gate actions

Codar allows you to configure release gate actions at each lifecycle stage. These actions act as a gate between two lifecycle stages. A package has to pass through each action that is defined; hence they are called release gate actions. During promotion of the package, the gate actions defined in each are executed. After all the actions complete successfully, the package transitions to the next lifecycle stage.

You can configure the following release gate actions:

- Deploy Application - this action deploys the Codar application design on the specified target environment. If the design is a partial design, the micro service designs that meet the requirements of the application design must also be selected.

- Run OO Flow - this action executes the specified flow in Operations Orchestration. Typically this action can be used to execute specific tests, with or without a deployment instance.

- Add Approver - this action is used to configure approvers who will approve the package promotion. It is also used to configure the minimum number of approvals, automatic approvals, and the number of days after which the automatic approval process is triggered.

- Execute Test Set - this action can be used for executing specific tests with or without a deployment instance.

- Run Script - this action can be used to execute a script on a remote machine, with or without a deployment instance.

- Provision Cloud Service - this action can be used to execute cloud offerings from CSA. After an offering is realized, the same can be used to deploy applications.

Note: You need permission to configure release gate actions.

Deploy Design action versus Run OO Flow action

Deploy Design actions are typically created to test your environment and to perform a test deployment before promoting your design to the production stage.

Run OO Flow actions are typically used to execute operations or tests on a deployed instance. You can also execute these actions without an instance of a deployment.

Disable or Enable a release gate action

Any action you create will be in enabled state by default. You can disable or enable a disabled release gate action during any stage in a lifecycle.

To disable or enable a release gate action

1. Navigate to **Release Automation** > **Release Pipeline**.

2. Select the design type (Topology Designs or Sequenced Designs) from the drop-down list.

3. Select the application and the version which contains the release gate action you want to disable or enable.

4. Click the gear icon next to the lifecycle stage which contains the release gate action you want to disable or enable and click **Configure Gate Actions**. The **Configure Gate Actions for <*lifecycle stage*** dialog appears, listing the available release gate actions.

5. Use the **Enable** or **Disable** link appearing against the respective release gate action to enable or disable the same.

> **Note:** You cannot disable or enable release gate actions, if there is a dependency among the release gate actions being disabled/enabled .
>
> Similarly, you cannot rearrange the order of execution of release gates in case there is a dependency among the release gates being rearranged.
>
> For example, if output properties of Release Gate action 'A' have been used in Release Gate Action 'B', then you cannot rearrange action 'B' to be executed before action 'A'.

Create a deploy application release gate action

You can create a deploy application release gate action during any stage in a lifecycle.

To create a deploy application release gate action:

1. Navigate to **Release Automation** > **Release Pipeline**.

2. Select the design type (Sequenced Designs or Topology Designs) from the drop-down list.

3. Select the application and the version for which you want to create a release gate action.

4. Click the gear icon next to the lifecycle stage in which you want to create the action and click **Configure Gate Actions**.

5. Click **Add Action** > **Deploy Application** to create a deploy action.

6. In the Deploy Action dialog box, fill in the following fields:

Field	Description
Display Name	Name of the deploy application action. This is displayed automatically. You can change it, if required.
Description	Description of the action you are creating
Environment	Select the environment in which you want to create the deployment.

7. In the Success and Failure Configuration area, select the check boxes as applicable:

Check box	Description
Notify on Success	Select this check box for a notification message to be sent if the deploy action executes successfully. The notification message is sent to the user who initiated the deploy action.
Notify on Failure	Select this check box for a notification message to be sent if the deploy action fails to execute successfully. The notification message is sent to the user who initiated the promotion request.
Reject on Failure (Blocking)	Select this check box to reject the package if the deploy action fails to execute successfully.
Cleanup on Promote Success	Select this check box to undeploy or clean up the deployed instances if the package is successfully

Check box	Description
	promoted to the next stage.
Cleanup on Promote Failure	Select this check box to undeploy or clean up or remove the deployed instance if the package has failed to promote to the next stage.

8. (Optional) Click **Next** and change/specify modifiable properties for this release gate action.

   For topology designs, default values of modifiable properties will be displayed automatically. You can change these values, if required.

   For sequenced designs, all properties with default value are displayed automatically.

9. Click **Deploy**.

   The action is listed in the 'Configure Gate Actions for <lifecycle stage>' dialog box.

Edit or delete a deploy application release gate action

You can edit or delete the details of a release gate action during any stage in a lifecycle.

Edit a release gate action

To edit a release gate action:

1. In the **Configure Gate Actions for <*lifecycle stage*** dialog box, click the gear icon next to the release gate action that you want to edit and click **Edit**. **Otherwise**, In the package workflow screen, sort the workflow by **View by Gate Actions** and click the release gate action you need to edit. For more information about package workflow screen, see "View the package workflow" on page 297.

2. In the Deploy Application dialog box, edit the details of the action. You can edit the name, description, environment, select/deselect any of the check boxes in the Success and Failure Configuration area, or change modifiable properties. For information about these fields, see Create a deploy application release gate action.

3. Click **Deploy**.

   The action has been edited.

Delete a release gate action

To delete a release gate action:

1. In the **Configure Gate Actions for <*lifecycle stage*** dialog box, click the gear icon next to release gate action that you want to delete and click **Delete**.

2. Click **Yes** in the confirmation box.

> If the deploy application action is associated with an OO flow action or a test set action, you cannot delete the deploy application action without first deleting the OO flow / test set action.

The action has been deleted.

Create an Operations Orchestration flow release gate action

You can create an Operations Orchestration (OO) flow release gate action at any stage in a lifecycle.

To create an OO flow release gate action:

1. Navigate to **Release Automation** > **Release Pipeline**.

2. Select the design type (sequenced or topology) from the drop-down list in the left pane. (Sequenced Designs is selected by default).

3. Select the application and the version where you want to create an OO flow release gate action.

4. Click the gear icon next to the lifecycle stage in which you want to create the OO flow action and click **Configure Gate Actions**.

5. Click **Add Action** > **Run OO Flow** to create an OO flow action.

6. a. In the Run OO Flow dialog box on the General tab, fill in the following fields:

Field	Description
Display Name	Name of the OO flow action. This is displayed automatically. You can change it, if required.
Description	Description of the action you are creating
Select Provisioning	Optionally, select the Deploy Application action or the Provision Cloud Service action that you want to associate with the OO flow action. All the Deploy Application and Provision Cloud Service actions that have been created for the selected package are displayed. **Note**: If you do not select a deployment or Provision Cloud Service action here, then you will be able to map a constant value or a context property in the **Parameters** tab. Whereas if you select a deploy or Provision Cloud Service action, you will have an additional option to map a component property.

b. In the Success and Failure Configuration area, select the check boxes as applicable:

Check box	Description
Notify on Success	Select this check box for a notification message to be sent if the OO flow action executes successfully. The notification message is sent to the user who initiated the promotion request.
Notify on Failure	Select this check box to receive a notification email if the OO flow action fails to execute successfully. The notification message is sent to the user who initiated the promotion request.
Reject on Failure (Blocking)	Select this check box to reject the package if the OO flow action fails to execute successfully.
Ignore on Failure	Select the check box to mark the specified action optional (non-blocking). The execution will move to the next release gate action even if the action fails to execute successfully.

> You can select more than one check box. However, for the **Reject on Failure** and the **Ignore on Failure** options, you can select only one of them at the same time.

7. Click **Next**.

8. On the Run OO Flow tab, select an OO workflow for the OO flow action. The OO workflow defines the course of your OO flow action and is invoked by passing the relevant parameters during the execution of the OO flow action.

9. Click **Next**.

10. The Parameters tab displays the name and path of the OO workflow, and the list of parameters present in it. You need to map these parameters to the context property or constant value present in the design. To map the parameters:

   a. Select the gear icon next to the parameter you want to map to and click **Edit**.

   b. In the Edit Parameter Mapping dialog box, fill in the following fields:

Field	Description
Display Name	Name of the parameter. You can change the default name to a name of your choice.
Description	Description of the parameter
Mapping Type	Object that you want to map the parameter to. You can either map the parameter to a

Field	Description
	component property, context property or constant value.
	If you select Context Property as the Mapping Type, you must select any of the following Context properties from the **Value** field.
	If you select Constant Value as the Mapping Type, you ,must provide a constant value in the **Value** field.
	For a password parameter, if you set the **Mapping Type** as **Constant Value**, then the password that you type in the **Value** field will be masked. This ensures the safety of your password.
	If a deploy or Provision Cloud Service action is not associated with this OO flow action, then you can map either a constant value or a context property only. Whereas if a deploy or Provision Cloud Service action is associated with it, then you will have an additional option to map a component property.
	Select 'Release Gate Action' to map the output parameters of one release gate action as input parameters to the next release gate action.
	Note: The release gate action being mapped to another release gate action must be such that it generates **output parameters** upon successful execution.
Components	Displayed only if you selected Component Property or Package Property in the **Mapping Type** field. Select the component that you want to map the parameter to. All the components that are displayed are a part of the Codar design.
Component Property	Displayed only if you selected Component Property or Package Property in the

Field	Description
	Mapping Type field. Select the component property for the component you selected. This is a dynamic list whose values change depending on the component you selected. The value of the property is obtained from the deployed instance/Cloud Service selected for this OO flow action. This option only lists concrete components for partial designs. It does not include any server capability components. **Note:** For **Mapping Type** as 'Package Property', the system picks up the updated value of a package property for the selected Component.
Gate Actions	Displayed only if you selected Release Gate Actions in the **Mapping Type** field. Select the gate action you want to map to this gate action.
Gate Action Properties	Displayed only if you selected Release Gate Actions in the **Mapping Type** field. Select the gate action property to be mapped to this gate action.

Mapping of parameters marked with an asterisk is mandatory. If you do not map such parameters, an error message is displayed when you try to add a OO flow action.

You can also unmap a parameter that is already mapped to a component. To unmap a parameter from a component, select the gear icon next to the parameter and click **Unmap**.

   c. Click **Save**.

11. Click **Add Action**.

The action that you added is listed in the **Configure Gate Actions for <lifecycle stage>** dialog box.

Note: After creating this OO flow release gate action, if any of the OO flow properties are modified or a new property is added using Operations Orchestration, those changes will be automatically ported to this release gate action.

Edit or delete an OO flow release gate action

You can edit or delete the details of an OO flow release gate action during any stage in a lifecycle.

Edit an OO flow release gate action

To edit an OO flow release gate action:

1. In the **Configure Gate Actions for <*lifecycle stage*>** dialog box, click the gear icon next to release gate action that you want to edit and click **Edit**.

2. In the Run OO Flow dialog box, edit the details of the action. You can edit the name, description, deployment, or select/deselect any of the check boxes in the Success and Failure Configuration area. For information about these fields, see "Create an Operations Orchestration flow release gate action" on page 266.

3. You can also edit the name, description, mapping type, components, or component type of any parameter.

4. Click **Save Action**.

   The action has been edited.

Delete an OO flow release gate action

To delete an OO flow release gate action:

1. In the Configure <*lifecycle stage*> Gate Actions dialog box, click the gear icon next to release gate action that you want to delete and click **Delete**.

2. Click **Yes** in the confirmation box.

The action is deleted.

Create an approval release gate action

An approval action acts as a final gate before you promote your package to the next lifecycle stage. You can choose the users who must approve the promotion to the next stage. Choosing users that belong to administrator or similar roles and granting the right to approve to a limited set of users minimize the chances of accidental promotion to the next stage.

When an approval action is executed, an email is sent to the approvers notifying them of a pending approval request. They can then approve or deny the promotion using the **Release Automation** > **Approvals** sidebar menu item.

When an approval action is executed, an email is sent to the approvers notifying them of a pending approval request.

To create an approval action

1. Navigate to **Release Automation** >> **Release Pipeline**.

2. Select the design type (sequenced or topology) from the drop-down list in the left pane. (Sequenced Designs is selected by default).

3. Select the application and the version for which you want to create an approval action.

4. Click the gear icon next to the stage where you want to create an approval action, and click **Configure Gate Actions**.

5. Click **Add Action** > **Add Approver** to create an approval action.

6. In the Add Approval Policy dialog box:

   a. In the **Display Name** filed, enter the name of the approval action. This is displayed automatically. You can change it, if required.

   b. In the **Description** filed, enter a description for the action you are creating.

   c. In the **Assigned Approvers** field, click the + button to search for and add approvers who must approve the promotion. When you click the + button, the Add Users dialog box is displayed.

   d. In the Add Users dialog box, the drop-down list displays all the roles that are present in the design. You can select a particular role and then type the user name or the starting alphabets of the user name that belongs to the role.

   > Only user names configured in LDAP are displayed in the drop-down list.

e. Click **Search**. All user names that belong to the role you selected are displayed. If you are not sure which role the user name belongs to, select **All Roles**. You can select multiple user names.

f. In the **Minimum Approvals/Rejections** field, enter the minimum number of approvers who must approve or reject the promotions. For example, if you enter 2, at least 2 of the approvers you added in the **Assigned Approvers** field must approve for the package to be promoted to the next lifecycle stage.

g. Select the **Automatic Approval** check box for the package to be automatically promoted or denied if the assigned approvers do not approve or reject within the hours specified in the **Wait time for Automatic Approval (in hours)** field.

h. In the **Automatic Approve/Reject** drop-down list, select whether the package must be approved or rejected during the approval process.

For example, if the minimum number of approvers is 3, and you select **Approved** from the **Automatic Approve/Reject** drop-down list and specify a wait time of 2 hours, then even if only 1 approver approves and the other 2 approvers do not respond, the package is automatically promoted to the next lifecycle stage after 2 hours.

> The **Automatic Approve/Deny** and **Wait time for Automatic Approval** fields are displayed only if the **Automatic Approval** check box is selected.

i. In the **Wait time for Automatic Approval (in hours)** field, enter the number of hours after which the package is automatically promoted to the next lifecycle stage.

j. Click **Add**.

An email is now sent to all the approvers named in the **Assigned Approvers** field. Approvers have the right to "Approve or deny an approval request" on page 315.

Edit or delete an approval release gate action

You can edit or delete the details of an approval release gate action during any stage in a lifecycle.

Edit an approve release gate action

To edit an approve release gate action:

1. In the **Configure Gate Actions for <*lifecycle stage*>** dialog box, click the gear icon next to release gate action that you want to edit and click **Edit**.

2. In the Edit Approver dialog box, edit the details of the action. You can edit the name, description, assigned approvers, and so on. For information about these fields, see "Create an approval release gate action" on page 271.

3. Click **Save Action**.

   The action has been edited.

Delete an approve release gate action

To delete an approve release gate action:

1. In the Configure <*lifecycle stage*> Gate Actions dialog box, click the gear icon next to release gate action that you want to delete and click **Delete**.

2. Click **Yes** in the confirmation box.

The action is deleted.

Create a test set release gate action

You can create a test set release gate action at any stage in a lifecycle.

To create a test set release gate action:

1. Navigate to **Release Automation** > **Release Pipeline**.

2. Select the design type (sequenced or topology) from the drop-down list in the left pane. (**Sequenced Designs** is selected by default).

3. Select the application and the version for which you want to create a test set release gate action.

4. Click the gear icon next to the stage where you want to create the test set action and click **Configure Gate Actions**.

5. Click **Add Action** > **Execute Test Set** to create a test set action.

6. In the **Execute Test Set** dialog box, in **General** tab, fill in the following fields:

Field	Description
Display Name	Name of the test set action. This is displayed automatically. You can change it, if required.
Description	Description of the test set action
Select Deployment	Select the deploy action that you need to associate with the test set action. All the deploy actions that have been created for the selected package are displayed. If you do not select a deploy action here, then you will be able to map a constant value or a context property in the **Parameters** tab. Whereas if you select a deploy action, you will have an additional option to map a component property.

In the Success and Failure Configuration area, select the check boxes as applicable:

Check box	Description
Notify on Success	Select this check box for a notification message to be sent if the test set action executes successfully. The notification message is sent to the user who initiated the promotion request.
Notify on Failure	Select this check box to receive a notification email if the test set action fails to execute successfully. The notification message is sent to the user who initiated the promotion request.

Check box	Description
Reject on Failure (Blocking)	Select this check box to reject the package if the test set action fails to execute successfully.
Ignore on Failure	Select the check box to mark the specified action optional (non-blocking). The execution will move to the next release gate action even if the action fails to execute successfully.

You can select more than one check box. However, for the **Reject on Failure** and the **Ignore on Failure** options, you can select only one of them at the same time.

7. Click **Next**.

   a. In the **Execute Test Set** dialog box, select the ALM provider instance.

      If only one ALM provider instance is configured in Codar, then it is loaded by default. Whereas, if more than one ALM provider instances are configured, you need to select one. Codar lists the test sets available in the selected ALM provider instance.

      Note: You can create test set action only if you have an ALM provider instance with right configurations.

      In the list, you will find all types of test sets, i.e., Performance, Default and Functional test sets that are available in the selected ALM provider instance. However, test set action execution is supported only for Functional test sets. So, while creating a test set action, you must select functional test sets only.

   b. At this point, you can decide whether to proceed by associating an AUT environment with the test set action, or not.

      To map an AUT environment with this test set action, check **Selected Test Set requires AUT Environment parameters** check-box. Click **Next** button. You are navigated to the next step where you can select the AUT Environment.

      In order to proceed without mapping an AUT environment, uncheck **Selected Test Set requires AUT Environment parameters** check-box. You can skip AUT Environment selection step and directly click **Add Action** button to add the test set action.

8. Click **Next**.

9. The **Parameters** dialog box displays the list of input parameters present in the environment. You need to map these parameters to the context property or constant value present in the design. To

map the parameters:

a. Select the gear icon next to the parameter you want to map to and click **Edit**.

b. In the **Edit Parameter Mapping** dialog box, fill in the following fields:

Field	Description
Display Name	Name of the parameter. You can change the default name to a name of your choice.
Description	Description of the parameter
Mapping Type	Object that you want to map the parameter to. You can either map the parameter to a component property, context property or constant value. If you select Context Property as the Mapping Type, you must select any of the following Context properties from the **Value** field. If you select Constant Value as the Mapping Type, you ,must provide a constant value in the **Value** field. If you have selected a deploy action in the **General** tab, then you can also select mapping type Component Property. If you select Component Property, then you must also select the components and the component property. > If a deploy action is not associated with this test set action, then you can map a either a constant value or a context property only. Whereas if a deploy action is associated with it, then you will have an additional option to map a component property. Select 'Release Gate Action' to map the output parameters of one release gate action as input parameters to the next release gate action. **Note:** The release gate action being mapped to another release gate action must be such that it generates **output**

Field	Description
	parameters upon successful execution.
Components	Displayed only if you selected Component Property or Package Property in the **Mapping Type** field.
	Select the component that you want to map the parameter to. All the components that are displayed are a part of the Codar design.
Component Property	Displayed only if you selected Component Property or Package Property in the **Mapping Type** field.
	Select the component property for the component you selected. This is a dynamic list whose values change depending on the component you selected. The value of the property is obtained from the deployed instance selected for this test set action.
	This option only lists concrete components for partial designs.
	Note: For **Mapping Type** as 'Package Property', the system picks up the updated value of a package property for the selected Component.
Gate Actions	Displayed only if you selected Release Gate Actions in the **Mapping Type** field.
	Select the gate action you want to map to this gate action.
Gate Action Properties	Displayed only if you selected Release Gate Actions in the **Mapping Type** field.
	Select the gate action property to be mapped to this gate action.

Mapping of parameters marked with an asterisk is mandatory. If you do not map such parameters, an error message is displayed when you try to add a test set action.

You can also unmap a parameter that is already mapped to a component. To unmap a parameter from a component, select the gear icon next to the parameter and click **Unmap**.

  a. Click **Save**.

10. Click **Add Action**.

The action that you added is listed in the 'Configure Gate Actions for <lifecycle stage> ' dialog box.

Edit or delete a test set release gate action

You can edit or delete the details of a test set release gate action during any stage in a lifecycle.

Edit a test set release gate action

To edit a test set release gate action:

1. In the **Configure Gate Actions for** *<lifecycle stage>* dialog box, click the gear icon next to release gate action that you want to edit and click **Edit**.

2. In the **Execute Test Set** dialog box, edit the details of the action. You can edit the name, description, deploy action and the success and failure configurations. For information about these fields, see "Create a test set release gate action" on page 274.

3. You can view the ALM provider, ALM test sets and AUT parameters selected for the test set action.

4. You can also edit the name, description, mapping type, components, or component type of any parameter. Parameters can be defined only if an AUT environment is selected for the test set action. If an AUT environment is not selected, then there will not be any parameters to view or edit.

5. Click **Save Action**.

   The action is updated.

Delete a test set release gate action

To delete a test set release gate action:

1. In the **Configure Gate Actions for** *<lifecycle stage* dialog box, click the gear icon next to release gate action that you want to delete and click **Delete**.

2. Click **Yes** in the confirmation box.

The action is deleted.

Create a script release gate action

You can create a script release gate action at any stage in a lifecycle, to execute scripts (shell scripts for Linux and Powershell for Windows) in the target machines during the promotion of a package.

Before creating a script release gate action, you must create a resource provider with **Provider Type** as **GitHub**. You can later associate this resource provider with a script release gate action. Refer to Provider overview for more information.

> **Note:** Ensure that appropriate values are provided in the following fields of the **Properties** tab when creating the GitHub resource provider:
>
> - **Access Token**: A system to authenticate GitHub REST API. Required only if the GitHub repository is private. Leave this field blank if the repository is public. You need to generate the personal access token under 'Settings' in your GitHub account at https://github.n/ or your GHE website, and specify the same here.
>
> - **Owner**: Owner of the repository. This is a mandatory field.
>
> - **Proxy**: The value required for communication between the Codar machine and GitHub repository server and later between the remote host machine and the GitHub repository server. Mandatory only if the repository needs to be accessed from the local machine.
>
> - **Repository URL** : Allows you to limit which content/script is exposed to the user, from the repository. For example, if you want to limit the access to a particular script in a folder, you just need to mention the path of that folder here. The user will not be able to access any script/content other than what is present in this folder.

> **Important:** Codar does not validate the script content. If there is any malicious data in the script, it might harm the system.
>
> For security reasons, it is recommended that you add an approval action before the script action to validate the contents and mitigate the risk.

To create a script release gate action:

1. Navigate to **Release Automation** > **Release Pipeline**.

2. Select the design type (sequenced or topology) from the drop-down list in the left pane. (**Sequenced Designs** is selected by default).

3. Select the application and the version for which you want to create a script release gate action.

4. Click the gear icon next to the stage where you want to create the script action and click **Configure Gate Actions**.

5. Click **Add Action** > **Run Script** to create a script action.

6. In the **Run Script** dialog box, in **General** tab, fill in the following fields:

Field	Description
Display Name	Name of the script action. This is displayed automatically. You can change it, if required.
Description	Description of the script action
Select Deployment	Select the deploy action that you need to associate with the script action. All the deploy actions that have been created for the selected package are displayed.
	If you do not select a deploy action here, then you will be able to map a constant value or a context property in the **Parameters** tab. Whereas if you select a deploy action, you will have an additional option to map a component property.

In the Success and Failure Configuration area, select the check boxes as applicable:

Check box	Description
Notify on Success	Select this check box for a notification message to be sent if the script action executes successfully. The notification message is sent to the user who initiated the promotion request.
Notify on Failure	Select this check box to receive a notification email if the script action fails to execute successfully. The notification message is sent to the user who initiated the promotion request.
Reject on Failure (Blocking)	Select this check box to reject the package if the script action fails to execute successfully.
Ignore on Failure	Select the check box to mark the specified action optional (non-blocking). The execution will move to the next

Check box	Description
	release gate action even if the action fails to execute successfully.

> You can select more than one check box. However, for the **Reject on Failure** and the **Ignore on Failure** options, you can select only one of them at the same time.

7. Click **Next**.

8. In **Host Details** tab, fill in the following details:

   a. **Time Out (in Seconds)**: Specify the maximum duration after which the script execution must time out.

   b. **OS Type**: Select the type of operating system on which the script is to be executed, from the drop-down list.

      Script actions under Linux OS type can run scripts and hence can be executed on Ubuntu, RHEL, etc.

      Script actions under Windows OS type can run powershell scripts on Windows platform such as Windows 2012.

      > **Note:** It is a prerequisite that script executions be allowed beforehand on the host machine being configured. For example, in case of a Windows host machine, the ExecutionPolicy should be set.

   c. **Host Details**

Field	Description
Host Name	Name of the host machine on which the script is to be executed.
Host Password	Host machine password.
Host Private Key	Host machine private key (applicable to Linux machines only).
Host User Name	Host machine user name.

   You can add or edit values in each of the above fields (parameters). To do so:

   a. Select the gear icon next to the parameter you want edit and click **Edit**.

   b. In the **Edit Parameter Mapping** dialog box, fill in the following fields:

Field	Description
Display Name	Name of the parameter. You cannot change the default name.
Description	Description of the parameter.
Mapping Type	Object that you want to map the parameter to. You can either map the parameter to a component property, context property or constant value.

If you select Context Property as the Mapping Type, you must select any of the following Context properties from the **Value** field.

If you select Constant Value as the Mapping Type, you ,must provide a constant value in the **Value** field.

If you have selected a deploy action in the **General** tab, then you can also select mapping type Component Property. If you select Component Property or Package Property, then you must also select the components and the component property.

Note: For Mapping Type as 'Package Property', the system picks up the updated value of a package property for the selected Component.

> If a deploy action is not associated with this script action, then you can map a either a constant value or a context property only. Whereas if a deploy action is associated with it, then you will have an additional option to map a component property.

Select 'Release Gate Action' to map the output parameters of one release gate action as input parameters to the next release gate action.

Note: The release gate action being mapped to another release gate action must be such that it generates **output parameters** upon successful execution. |
| Value | Select one of the following values:

- Source Lifecycle Stage
- Destination Lifecyle Stage
- Package Name
- Action Instance Id |

Field	Description
	• Request User
Gate Actions	Displayed only if you selected Release Gate Actions in the **Mapping Type** field. Select the gate action you want to map to this gate action.
Gate Action Properties	Displayed only if you selected Release Gate Actions in the **Mapping Type** field. Select the gate action property to be mapped to this gate action.

> Mapping of parameters marked with an asterisk is mandatory. If you do not map such parameters, an error message is displayed when you try to add a script action.

You can also unmap a parameter that is already mapped to a component. To unmap a parameter from a component, select the gear icon next to the parameter and click **Unmap**.

9. Click **Next**.

10. In the **Script Details** tab, fill in the following fields:

Field	Description
SCM Type	Select the GitHub provider through which you want to execute the script.
SCM Instance	Select the script instance. If there is an error in Provider configuration, an appropriate error message will be displayed and you cannot proceed further until the error is fixed.

11. Click **Next**.

12. In the **Script Arguments** tab, add the script arguments. To do so:
    a. Click the gear icon next to 'Argument0' and select 'Edit'.

    b. In the **Edit Parameter Mapping** dialog, specify the details as explained above for **Host Details** tab.

    c. Click **Save**.

    d. Click '+' icon and repeat procedure from 'a' to 'c' to add as many arguments as you need.

13. Click **Add Action**.

The script action that you added will be listed in the **Configure Gate Actions** dialog box.

> **Note:** Running script action (such as Windows PowerShell) on the target machine is not supported when both Codar and OO are installed on a non-Windows machine. To overcome this limitation, you must set up an OO RAS (Remote Action Services) server on a Windows machine, which will in turn run the script action on the target machine. Refer to 'Operations Orchestration Installation, Upgrade, and Configuration' guide for information on installing and configuring an OO RAS server.

How Codar handles concurrency issues: When multiple script actions are being executed on the same host, the downloaded script filename is appended with a random number. Hence, name of the file will be <git_file_name><random_number>.

In addition, the file will be removed from the system after a successful execution of the script. In case the script execution fails, the respective file will be retained for debugging purpose.

Edit or delete a script release gate action

You can edit or disable a script release gate action during any stage in a lifecycle.

Edit a script release gate action

To edit a script release gate action:

1. In the **Configure Gate Actions for <*lifecycle stage*** dialog box, click the gear icon next to the script release gate action that you want to edit and click **Edit**.

2. In the **Run Script** dialog box, edit the details of the action. You can edit the name, description, deploy action and the succes and failure configurations. For information about these fields, see "Create a script release gate action" on page 280.

3. You can also edit the name, description, mapping type, components, or component type of any parameter.

4. Click **Save Action**.

   The action is updated.

Delete a script release gate action

To delete a script release gate action:

1. In the **Configure Gate Actions for <*lifecycle stage*** dialog box, click the gear icon next to release gate action that you want to delete and click **Delete**.

2. Click **Yes** in the confirmation box.

The action is deleted.

Create a Provision Cloud Service release gate action

You can create a Provision Cloud Service release gate action at any stage in a lifecycle, to fulfill cloud offerings during the promotion of a package.

During the creation of a Provision Cloud Service action, the offerings are listed based on the providers configured.

By default, the offerings published in the local instance of Codar are available for fulfillment.

> **Note:** The 'Provision Cloud Service' release gate action is not supported on machines with Codar 1.80 and CMP Provider 4.80 or below.

Refer to Provider overview for more information.

To create a Provision Cloud Service release gate action:

1. Navigate to **Release Automation** > **Release Pipeline**.

2. Select the design type (sequenced or topology) from the drop-down list in the left pane. (**Sequenced Designs** is selected by default).

3. Select the application and the version for which you want to create a Cloud Service release gate action.

4. Click the gear icon next to the stage where you want to create the offerings action and click **Configure Gate Actions**.

5. Click **Add Action** > **Provision Cloud Service**.

6. In the **Provision Cloud Service** dialog box, in **General** tab, fill in the following fields:

Field	Description
Display Name	Name of the Provision Cloud Service action. This is displayed automatically. You can change it, if required.
Description	Description of the Provision Cloud Service action.
Time Out (in Seconds)	Specify the time after which the API call must time out if the execution does not happen.
Subscription Duration (in Seconds)	Specify the duration (in terms of seconds) for which the Cloud Service is valid.

In the Success and Failure Configuration area, select the check boxes as applicable:

Check box	Description
Notify on Success	Select this check box for a notification message to be sent if the Provision Cloud Service action executes successfully. The notification message is sent to the user who initiated the promotion request.
Notify on Failure	Select this check box to receive a notification email if the Provision Cloud Service action fails to execute successfully. The notification message is sent to the user who initiated the promotion request.
Reject on Failure (Blocking)	Select this check box to reject the package if the Provision Cloud Service action fails to execute successfully.
Ignore on Failure	Select the check box to mark the specified action optional (non-blocking). The execution will move to the next release gate action even if the action fails to execute successfully.

> You can select more than one check box. However, for the **Reject on Failure** and the **Ignore on Failure** options, you can select only one of them at the same time.

7. Click **Next**.

8. In **Select Offering** tab, fill in the following details:

    a. **Offering Provider Instance**: Select the required provider instance from the drop-down list, if more than one provider instances are available.

    b. **Offerings**: Select the required offering, from the list. You can also search for the required offering using the **Search** button.

9. Click **Next**.

10. In **Offering Details** tab, based on the selected offering, provide the values for the listed parameters. This tab lists the parameters based on how the offering is created in Cloud Service (CSA offering).

11. Click **Add Action**.

The Provision Cloud Service action that you added will be listed in the **Configure Gate Actions for <lifecycle stage>** dialog box.

Edit or delete a Provision Cloud Service release gate action

You can edit or delete the details of a Provision Cloud Service release gate action during any stage in a lifecycle.

Edit a Provision Cloud Service release gate action

To edit a Provision Cloud Service release gate action:

1. In the **Configure Gate Actions for <*lifecycle stage*** dialog box, click the gear icon next to release gate action that you want to edit and click **Edit**.

2. In the **Provision Cloud Service** dialog box, edit the details of the action.

3. Click **Save Action**.

   The release gate action is updated.

Delete a Provision Cloud Service release gate action

To delete a Provision Cloud Service release gate action:

1. In the **Configure Gate Actions for <*lifecycle stage*** dialog box, click the gear icon next to release gate action that you want to delete and click **Delete**.

2. Click **Yes** in the confirmation box.

The release gate action is deleted.

Notifications

Different types of notifications are sent during the execution of release gate actions. They are:

- Stage transition notifications

  By default, notification messages are sent to all user groups added in the Manage Access Control dialog box of the application design when a package is promoted successfully from one lifecycle stage to another. This notification is sent only if user groups are added to the application design under access control.

- Action complete notification

  The notification is sent to the user who promoted the package. This notification is governed by the **Notify on Success** and **Notify on Failure** options available in each action.

- Approval request notification

  This notification is sent to approvers informing them of an approval request that is triggered during the execution of a release gate approval action.

View package statistics

The Pipeline Statistics tab displays detailed information about packages and includes graphical representations of package summary, trends, states, deployment status and so on. It provides a holistic view of all packages and deployments and enables you to make informed decisions with respect to package deployment.
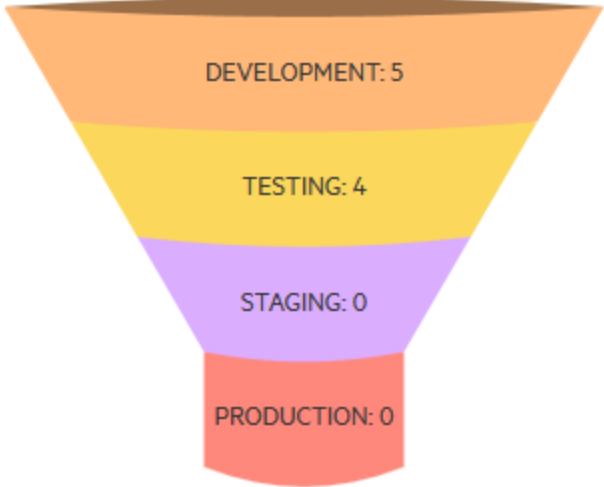
To view package statistics

1. Navigate to **Release Automation** > **Release Pipeline**.

2. Select the type of design (sequenced or topology) for which you need to view the package statistics, from the drop-down list on the left hand side of the screen. (**Sequenced Designs** is selected by default).

3. From the list of designs on the left hand side of the screen, select the design whose package information you want to view.

4. Click the **Pipeline Statistics** tab in the right pane.

   You can view the pipeline statistics for all versions of the package. By default, the current version statistics are displayed. To view the statistics for an older version, select the Application Version from the drop-down list available on the right-hand side of the screen.

   To view the latest package statistics, click the gear button on the right-hand side of the screen and select **Refresh**.

   The following table lists the metrics that comprise pipeline statistics:

Metric	Description
Package Trend	Displays in line graph format, the days during which packages were created. The X-axis represents the dates and the Y-axis represents the number of packages. Hover your cursor on the X-axis to view the number of packages created on any date.
Release Progression	The mouth or the top layer of the funnel represents the first lifecycle stage and the bottom layer of the funnel represents the last lifecycle stage. Each layer displays the total number of packages that are not only present in the stage but also those that have successfully transitioned to the next stage. The colors that represent each lifecycle stage in the funnel are the colors that were set when the lifecycle stage was created.

For example, in the figure above, a total of 5 packages were in the Development stage, out of which 4 were sent to the Testing stage. From the Testing stage, 0 were sent to the Staging stage and from there 0 were sent to the Production stage.

This implies that out of 5 packages, 4 are in the Testing stage and 1 is still in the Development stage.

Package Summary	Displays in pie chart format the number of packages in each lifecycle stage. The colors used for each lifecycle stage are the colors set when the lifecycle stage was created. The number of packages in each state is displayed below the pie chart.
Deployment Summary	Displays in pie chart format the number of deployments in each lifecycle stage. The colors used for each lifecycle stage are the colors set when the lifecycle stage was created. The number of deployments in each state is displayed below the pie chart.
Package Status	Displays in bar graph format, in each lifecycle stage the number of packages in each state. The X-axis represents the lifecycle stages and the Y-axis represents the number of packages. The different colors represent the different package states. Select **Grouped** to view different bars for each package state or **Stacked** to view all package states in one bar. See "Lifecycle stages and actions" on page 309 for details about package states.
Deployment	Displays in bar graph format, in each lifecycle stage the number of deployments

Status	in each state. The X-axis represents the lifecycle stages and the Y-axis represents the number of deployments. The different colors represent the different deployments.
	This graph depicts the number of deployments that are not canceled in the system and how many resources are consumed.
	Select **Grouped** to view different bars for each deployment or **Stacked** to view all deployments in one bar.

Access control

You can configure access control at user groups level for a sequenced design or a topology design.

- If a design is created by a user with the administrator role, then all users will have access to the design if you do not configure access control.

- If a design is created by a user with the Application Architect role, then only that user will have access to the design if you do not configure access control.

- If you add groups to access control, then only the users within those groups can access the design. Other users cannot see it.

Users with the Application Developer, Application QA, or Application Release Manager role have read-only access to designs and can view designs in the Release Pipeline area.

The groups are from an LDAP server and are configured in the Provider organization. See "Configure an organization" on page 1.

An external group is an LDAP group that includes users with different roles. The external groups are configured in LDAP. When the external group is added to access control, the users who are part of other groups configured in the organization can log in and access the design. Users who are not part of the organization cannot log in.

> **Note**: The admin user can create and access all designs, but is not listed in the Access Control tab.

To change access control for a design

1. Navigate to **Release Automation** > **Release Pipeline**

2. Select the type of design for which you need to configure access control, from the drop-down list on the left side of the screen. (**Sequenced Designs** is selected by default).

3. Select the design to manage access control.

4. Select **Access Control** tab for the design.

5. Click **Manage**.

6. Select either:

   Do not restrict access to this design to allow all users to access the design. This is the default.

> **Restrict access to this design to the following groups** and select the groups that will have access.

7. Add or remove groups if access control is enabled.

   To search for a group, enter the group name in the dialog. You can use an asterisk (*) as a wildcard when you search for a group. Use Shift or Ctrl to select multiple groups.

8. Click **Save**.

Tasks

You can perform the following tasks in the Access Control tab if you have the correct permission:

- **Sort the list of groups.** You can sort the list by selecting By Roles in the upper left side of the tab. When you sort by role, groups from LDAP are sorted into External Groups, while groups from the organization are sorted by role.

- **Search for groups.** Use the search box in the upper left side of the tab.

View access control for a design

The **Access Control** tab shows all roles that have been granted access to the design. Under the roles, it also displays user groups that belong to the role. This is a read-only view for all user roles except Application Architect role and can only be changed in the Designs area. See Access Control.

You can filter the page either by roles. If you filter by roles, all the roles are displayed and groups that belong to each role are displayed under that role.

View different versions for a single application design

The **Release Automation** > **Release Pipeline** >**Topology Designs or Sequenced Design**> **Versions** tab displays different versions of a selected application or sequenced design. It also displays all the lifecycle stages present in the design and the number of packages present in each stage. You can drill-down further and view the package overview and workflow by clicking a version number.

View the package workflow

The Package Workflow screen displays all the lifecycle stages in an application or sequenced design and the state of the packages in each lifecycle stage. To view the Package Workflow screen:

1. Navigate to **Release Automation** > **Release Pipeline**.

2. Select the type of design (sequenced or topology) for which you need to view the package workflow, from the drop-down list on the left side of the screen. (**Sequenced Designs** is selected by default).

3. Click the design and version for which you want to view the package workflow.

   The Package Workflow screen is displayed. This screen has several views:

   ○ View by Packages: Select **View by Packages** from the drop-down list on the right side of the screen to view all the packages in the design. Packages are displayed in their respective lifecycle stages. You can further filter by package states by selecting a package state from the drop-down list next to the **View by Packages** list.

   ○ View by Gate Actions: Select **View by Gate Actions** to view Release Gate Actions configured in respective lifecycle stages. You can view the Release Gate Actions in their respective lifecycle stage. Click an action to go to the **Edit Action** window.

   ○ View by Environments: Select **View by Environments** to view the selected Resource Environments in respective lifecycle stages.

   ○ View by Infrastructure Resource Pools: Select **View by Infrastructure Resource Pools** to view different resource pools configured across different lifecycle stages using the 'Configure Pools' option.

4. You can configure or create "Release gate actions" on page 261 for each lifecycle stage.

5. You can associate an environment with each lifecycle stage.

6. You can configure lifecycle stages by adding or removing a lifecycle stage. The lifecycle stage you add or remove applies only to the specific version you select. To configure lifecycle stages, select the gear icon on the right side of the screen and click **Configure Lifecycle Stages**.

Package statistics

The following statistics are displayed for each package available in the Package Workflow page.

- **Deployments Success:** The number of times the package has been deployed successfully.

- **Deployments Failed:** The number of times, the package deployment failed.

- **Deployments In progress:** The number of package deployments in progress currently.

- **Active/Transitioning/Rejected:** Package status, any one status displayed at a given time.

Associate an environment with a lifecycle stage

You can select resource environments that already exist and associate it with lifecycle stages. This helps you choose a subset of deployment environments to be associated with the lifecycle stage. By default all resource environments are associated with all lifecycle stages.

> **Note**: This option is only applicable to topology designs.

To select an environment to associate

1. Navigate to **Release Automation** > **Release Pipeline**.

2. From the list of topology designs on the left hand side, select the application design and version in which you select the environment.

3. Select the gear icon in the lifecycle stage with which you want to associate the environment and click **Select Environments**.

4. The Select Environments for <stage> dialog box displays all available resource environments. Select those that you want to associate and click **Add**. Selected environments are displayed in the Selected Environments box.

5. You can even search for environments using the Search fields. Type the name of the environment that you want to search for and press the Enter key.

6. Click **Save** to add the selected environments.

Configure purge schedules for packages and deployments

You can configure a schedule to automatically purge or remove packages and deployments periodically. This helps in freeing up of resources being held by packages/deployments that are no longer needed.

Users with 'Application Architect' or 'Admin' roles only can create or modify purge schedules.

You can perform the following tasks:

- "To view existing purge schedules" below

- "Create new purge schedules" on the next page

- "Edit existing purge schedules" on page 302

- "Delete existing purge schedules" on page 303

To view existing purge schedules

1. Go to **Release Automation** > **Release Pipeline**.

2. Select the design type (sequenced or topology) from the drop-down list in the left pane. (**Sequenced Designs** is selected by default).

3. Select the application design and version to view its purge schedule.

4. Click the gear icon corresponding to the lifecycle stage in which you need to configure the purge schedule, and select **Configure Purge Schedule**. The **Configure Purge Schedule for <lifecycle stage>** screen opens, displaying the list of purge schedules defined.

Create new purge schedules

To create a new deployment/package purge schedule

> **Note:** : Users with 'Application Architect' or 'Admin' roles only can create purge schedules.

1. Go to **Release Automation** > **Release Pipeline**.

2. Select the design type (sequenced or topology) from the drop-down list in the left pane. (**Sequenced Designs** is selected by default).

3. Select the application design and version in which you want to create a purge schedule.

4. Click the gear icon corresponding to the lifecycle stage in which you want to create the purge schedule, and select **Configure Purge Schedule**.

5. In the **Configure Purge Schedule for <lifecycle stage>** screen, click **Add Scheduler** and select the type of purge schedule you want to configure (deployment purge or package purge).

6. Specify the following details:

Field	Description
Display Name	Name of the purge schedule.
Purge <Deployments/Packages> Older than (in days)	Number of days after which deployments/packages in the given lifecycle stage must be purged.
Notification before (in days)	Number of days before which a notification must be sent about the impending purge.
Start Purge From	The date and time from which the schedule becomes active, in MM/DD/YYYY HH:MM AM/PM format. **Note:** Purge schedule date cannot be a date in the past. For a lifecycle stage, only one purge package schedule and one purge deployment schedule each can be created.

7. Click **Add Scheduler**.

8. Click **Done** to exit the **Configure Purge Schedule for <lifecycle stage>** screen.

Edit existing purge schedules

To edit an existing purge schedule

> **Note:** : Users with 'Application Architect' or 'Admin' roles only can modify purge schedules.

1. Go to **Release Automation** > **Release Pipeline**.

2. Select the design type (sequenced or topology) from the drop-down list in the left pane. (**Sequenced Designs** is selected by default).

3. Select the application design and version which contains the purge schedule you want to edit.

4. Click the gear icon corresponding to the lifecycle stage which contains the purge schedule you want to edit, and select **Configure Purge Schedule**.

5. Click the gear icon corresponding to the purge schedule you want to edit and select **Edit**.

6. Make the required changes and click **Save**.

7. Click **Done** to exit the **Configure Purge Schedule for <lifecycle stage>** screen.

Delete existing purge schedules

Only users with 'Admin' role or the schedule creator can delete purge schedules.

To delete an existing purge schedule

1. Go to **Release Automation** > **Release Pipeline**.

2. Select the design type (sequenced or topology) from the drop-down list in the left pane.
   (**Sequenced Designs** is selected by default).

3. Select the application design and version that contains the purge schedule you want to delete.

4. Click the gear icon corresponding to the lifecycle stage that contains the purge schedule you want
   to delete, and select **Configure Purge Schedule**.

5. Click the gear icon corresponding to the purge schedule you want to delete and select **Delete**.

6. Click **Yes** to confirm deletion.

7. Click **Done** to exit the **Configure Purge Schedule for <lifecycle stage>** screen.

View and cancel deployments

You can view all the deployments that you triggered during any lifecycle stage and choose to remove any of those deployments that are no longer relevant.

To view and remove deployments

1. Navigate to **Release Automation** > **Release Pipeline**.

2. Select the desired application design type (sequenced or topology), from the drop-down list on the left side of the screen. (**Sequenced Designs** is selected by default).

3. Select the application design and the version for which you want to view the deployments.

4. Click the **Deployments** tab.

   All the deployments that you triggered are displayed in a tabular format along with details such as the packages in each deployment, the lifecycle stage in which the deployment was triggered, and the status of the deployment.

5. You can filter the deployments based on lifecycle stages to view deployments pertaining to a specific lifecycle stage.

6. To remove a deployment, click the gear icon and click **Cancel**.

   All the packages in the deployment are listed in the Cancel Deployments dialog box. You can choose to remove one package or multiple packages.

   When you try to cancel a deployment that has a custom action associated with a deploy action, a warning message is displayed stating that upon cancellation, the custom action may not work. You can proceed to cancel the deployment by clicking 'Yes' in the message dialog box.

   > To delete a deployment, you must first cancel it.

7. Click **Cancel Deployments** and then **Yes** in the Confirmation Required confirmation box.

8. Click the gear icon and click **Delete**.

9. Select the packages to delete and click **Delete Deployments** in the Delete Deployments dialog box.

Note: You can cancel the package deployment of another user if you have the 'Cancel and Delete Deployments' permission, provided the deployment is being canceled in the same lifecycle stage in which it was created. Example: If a user, who has Development role associated with Development lifecycle stage, deploys a package, another user with 'Cancel and Delete

Deployments' permission can cancel that deployment in the Development lifecycle stage itself.

Configure lifecycle stages

You can configure lifecycle stages by adding or removing a lifecycle stage to the existing subset of lifecycle stages. The lifecycle stage you add or remove applies only to the specific version you select.

To configure lifecycle stages:

1. Go to **Release Automation** > **Release Pipeline**.

2. Select the design type (sequenced or topology) from the drop-down list in the left pane. (Sequenced Designs is selected by default).

3. Select the application design and version for which you want to configure the lifecycle stages.

4. Select the gear icon on the right side of the screen and click **Configure Lifecycle Stages**.

5. Click **Add Lifecycle Stage** to add a lifecycle stage to the existing set of lifecycle stages.

6. To remove a lifecycle stage, click the bin icon next to the lifecycle stage that you want to remove.

7. To change the order of the lifecycle stages, drag and drop them above or below each other in the order you want.

8. Click **Save**.

View lifecycle stage promotion schedules

You can view a list of all lifecycle stage promotions scheduled for packages of a particular application version, from the **Schedules** tab. You can view month or day-wise package promotion schedules.

To view package promote schedules

1. Click the **Release Pipeline** sidebar menu item.

2. Select the type of application design in which you need to view the promotion schedule, from the drop-down list on the left hand side (**Sequenced Designs** is selected by default).

3. Click the version of the application design which contains packages scheduled for promotion.

4. Click the **Schedules** tab.

    By default, the application displays a calendar view of month-wise schedules, showing dates on which a promote has been scheduled along with schedule name and time of schedule, for the month. You can switch between month and day views.

Pipeline Configurations

Use the Pipeline Configurations sidebar menu item to create and configure lifecycle stages and roles. You can also assign permissions to the roles.

Lifecycle stages and actions

Every lifecycle should comprises several stages and every stage has roles associated with it. It means that all users who belong to the roles in a particular lifecycle stage can perform operations defined in the role during that stage. For example, if the Development lifecycle stage has the Application Architect role associated with it, then users belonging to the Application Architect role can perform tasks as per the assigned permissions.

The following are the out-of-the-box lifecycle stages available in Codar:

- Development: This is usually the first stage in which the code is developed and application artifacts are created.

- Testing: Stage in which test cases are executed against the code developed in the Development stage.

- Staging: Pre-production stage that replicates the production environment; used to test the code and artifacts.

- Production: This is usually the final stage in which the application is deployed in a live environment.

Apart from the out-of-the-box lifecycle stages, there are custom stages that you can create. For information about creating, editing, and deleting custom stages, see "Create a lifecycle stage" on page 311 and "Edit or delete a lifecycle stage" on page 312.

The following table lists the actions pertaining to a package that take place in each lifecycle stage:

Stage	Promote	Deploy, Redeploy	Edit	Delete	Reject
First stage (usually the Development stage)	Yes	Yes	Yes	Yes	Yes
Intermediate	Yes	Yes	Yes	Yes	Yes
Final stage (usually the Production stage)	No	Yes	Yes	Yes	Yes

You can access lifecycle stages by using the **Release Automation** > **Pipeline Configurations** sidebar menu item.

Use the following actions to deploy or move a package through the stages. These actions describe the flow when release gate actions are not defined.

- **Promote**: Moves the package to the next lifecycle stage. The package state remains Active.

- **Deploy, Redeploy**: Deploys the package. See "Deploy a package" on page 236.

- **Edit**: Change the properties of a package. See "Edit a package" on page 234.

- **Reject**: Stops the package from advancing to another stage. The package will remain in its current stage, its state will be set to Rejected, and the action buttons will no longer be available.

- **Delete**: Delete a package. The package will be removed permanently from the system. A package can only be deleted if all associated deployed instances are canceled and deleted.

- **Refresh**: Retrieves current package status.

Package states

Packages have the following states:

- Active

- Rejected

- Transitioned

- Failed

If you reject a package, then it remains in its current stage, its state is set to Rejected, and no further actions can be applied; however, it can be deleted and will be removed from the system.

When a package is promoted, it moves to the next stage and remains in the active state. Packages are always created in the first stage. If the Codar Jenkins plug-in is configured, then after a successful build the Jenkins plug-in talks to Codar and creates a package.

Grouping service designs by lifecycle stage

A partial design with an active package requires a selection of a service design for provisioning in the deploy package wizard. These service designs can be grouped for different lifecycle stages so that the package deployment in the lifecycle stage lists only those grouped service designs from the respective lifecycle stage.

To group the service designs for a lifecycle stage, create a tag with the name of the lifecycle stage in each topology design. For example, you can create a Development tag and associate it with all required designs in the Development lifecycle stage.

The Test Run wizard in the Test lab lists all the designs and does not group by tag.

Create a lifecycle stage

Apart from the "Lifecycle stages and actions" on page 309 available in Codar, you can also create your own lifecycle stage (referred to as a custom lifecycle stage in this document) and associate both custom and out-of-the box roles with the lifecycle stage.

To create a custom lifecycle stage

Use the Pipeline Configurations sidebar menu item to create a custom lifecycle stage.

1. Navigate to **Release Automation** > **Pipeline Configurations** > **Lifecycle Stages**.

2. Select the gear icon next to **Lifecycle Stages** and click **Create Lifecycle Stage**.

3. In the Create Lifecycle Stage dialog, fill in the following fields:

Field	Description
Display Name	Name of the lifecycle stage
Description	Description of the lifecycle stage
Roles	Roles that must be associated with the lifecycle stage. Custom roles if any are also displayed.
Image	Select an image that can serve as an icon for the lifecycle stage.
Color	Select a color for the lifecycle stage. The color is useful if you want the differentiate various lifecycle stages on the basis of color.

4. Click **Create**.

The lifecycle stage is listed in the Lifecycle Stages page along with the roles associated with it.

Edit or delete a lifecycle stage

You can edit or delete not only custom lifecycle stages but also the out-of-the-box lifecycle stages available in Codar.

To edit a custom lifecycle stage

1. Navigate to **Release Automation** > **Pipeline Configurations** > **Lifecycle Stages**.

2. Select the gear icon next to the lifecycle stage you want to update and click **Edit**.

3. In the Edit Lifecycle Stage dialog, you can edit the name, description, icon, or color of the lifecycle stage. You can also add or remove roles associated with the lifecycle stage.

4. Click **Save**.

The edits are reflected in the Lifecycle Stages page.

To delete a custom lifecycle stage

1. Navigate to **Release Automation** > **Pipeline Configurations** > **Lifecycle Stages**.

2. Select the gear icon next to the lifecycle stage you want to update and click **Delete**.

3. Click **Yes** in the Confirmation Required dialog to delete the lifecycle stage.

The lifecycle stage is deleted.

> If the lifecycle stage that you are deleting is associated with a design, then when deleting the lifecycle stage, an error message is displayed.

Set default lifecycle stages for pipeline management

You can select lifecycle stages (both out-of-the-box and custom) and group these stages together to create a default or defined lifecycle stage. You can then apply the default lifecycle stage to a version of an application design.

Advantages of setting a default lifecycle stage

The main advantage of setting a default set of lifecycle stages to a new design version is that you do not need to manually define a lifecycle flow each time a new version of an application design is created.

To set default lifecycle stages

Use the Pipeline Configurations sidebar menu item to set the default lifecycle stage.

1. Navigate to **Release Automation** > **Pipeline Configurations** > **Lifecycle Stages**.

2. Select the gear icon next to **Lifecycle Stages** and click **Set Default Lifecycle Stage**.

3. When you log on to Codar for the first time, all out-of-the-box lifecycle stages are displayed. As and when custom lifecycle stages are created, those stages are also displayed. You can remove any lifecycle stage that you do not want to include as part of the default by clicking the bin icon at the right of the each lifecycle stage.

4. Click **Add Lifecycle Stage** and select the lifecycle stages that you want to be a part of the default set.

5. Click **Save**.

View all applications in a lifecycle stage

To view all application designs that belong to a particular lifecycle stage:

1. Navigate to **Release Automation** > **Pipeline Configurations**.

2. In the Lifecycle Stages page, click the lifecycle stage for which you want to see all the application designs.

   All the application designs that are in the lifecycle stage are displayed along with all their versions.

3. To remove a lifecycle stage from an application design, select the gear icon next to the lifecycle stage and click **Remove Lifecycle Stage**.

4. Click **Done** in the Remove Lifecycle Stage dialog box.

   The application design is removed from the lifecycle stage.

Approve or deny an approval request

When an approval request is created, an email is sent to the assigned approvers who can then approve or deny the promotion of a package to the next lifecycle stage.

To approve or deny an approval request

1. Navigate to **Release Automation** > **Approvals**.

   The **Approvals** sidebar menu item is displayed only to assigned approvers or users who have the "Application release gate approve" permission.

2. Click **Pending Requests** on the left panel. A list of all requests that require your approval is displayed.

   You can click on any request to drill-down and view the details of the request.

3. Select a request to approve or reject. You can also select multiple requests.

4. Click the **Approve** button to approve the request or the **Reject** button to reject it.

5. If you click the **Approve** button, entering a comment in the conformation box is optional but if you click the **Reject** button, you must mandatorily enter a comment.

6. Depending on whether you approve or reject a request, the request is moved to the **Approved Requests** area or the **Rejected Requests** area.

Configure servers

The **Release Automation** > **Server Configurations** > **Server Configuration** option displays existing servers. You can use this option for application deployment.

About existing server components

The existing server component is available out of the box with Codar. This component has the following properties:

- instanceId - Service instance ID. This is hidden and populated automatically.

- ipAddress - IP Address/ host name of existing server. This should always be represented by an existing server in Codar. Since the IP Address is not validated at the time of adding a server, you must add valid machines with valid credentials. The ipAddress is mandatory if the server is used for Test Run calls. The servers associated to this ipAddress should not be tagged to a non-shared pool. For Codar Pipeline use cases, if the ipAddress is specified for this component at the design level, then the server belonging to this ipAddress should be part of the associated pools for the package. If no pools are associated, then the server should not be associated to a non-shared pool.

- isBlocked - This is a flag to govern multiple application deployment on a server. By default, this option is set to false. When the option is set to false, there is a possibility of multiple applications being deployed on the same server (if no eligible server with zero deployment exists).

  Setting it to true will enforce that a single application be deployed on the server. If there are no eligible machines with zero prior deployments, then the deployment fails. You can change this property setting by performing a 'Save As' of the original component.

This component always tries to pick up the oldest least used server. The least used count refers to the number of times a server can be used for deployments. If the 'isBlocked' property is set to true at the component level, then always the oldest unused server will be picked up.

Test run for Design Scaling is not supported with this component.

Tasks

- View existing server configurations

- Add server components

- "Modify existing server components" on page 319

- "Delete existing server components" on page 320

View existing server configurations

To view existing server configurations:

1. Navigate to **Release Automation** > **Server Configurations**.

2. Click **Server Configuration** in the left pane.

The system lists all the available server configurations in the right pane.

Use the **Search** field to search for a specific server. You can search using any of the following attributes:

- Display name
- IP address
- host name

Server overview

Click any of the listed servers to view details such as name (combination of package name and application deployment name) deployed on this server, application name, package name, user who deployed the application, and the lifecycle stage(s) of the application.

Modify existing server components

To modify existing server components:

1. Navigate to **Release Automation** > **Server Configurations**.

2. In the **Server Configuration** page, click **Server Configuration** in the left pane. The system lists all the existing server components in the right pane.

3. Click the gear next to the server component you need to modify and select **Edit**.

4. In the **Edit Server** dialog, make the required modifications.

5. Click **Save**.

Delete existing server components

To delete existing server components:

1. Navigate to **Release Automation** > **Server Configurations**. The system lists all the existing servers in the right pane.

2. Click the gear next to the server component you need to delete and select **Delete**.

3. In the **Confirmation Required** dialog, click **Yes** to confirm deletion.

Associate pools

The **Release Automation** > **Server Configurations** > **Pools** page displays existing pools. Here, you can create pools so that they can be associated with lifecycle stages.

Tasks

- "View existing pools" below

- Create pools

- Modify existing pools

- Delete existing pools

View existing pools

To view existing pools:

1. Navigate to **Release Automation** > **Server Configurations**>**Pools**.

The **Pools** page is displayed, listing all the existing pools in the right pane.

Pools Overview

Click any of the listed pools to view details such as the application design with which it is associated, the version of the associated application, and the lifecycle stage(s) of the application.

Create pools

To create pools:

1. Navigate to **Release Automation** > **Server Configurations** > **Pools**.

2. In the **Pools** page, click the gear in the right pane and select **Create Pool**.

3. In the 'Create Pool' dialog, specify the following details:

Field	Description
Display name	Name of the pool.
Description	A brief description of the pool.
Server Configuration	You can associate existing servers to the pool being created. **Associating existing servers:** 1. Click the ⎘ icon to associate existing servers. 2. In the **Add New Server** dialog, select the server you want to associate. You can associate a single or multiple servers to the pool. Use the **Search** field to search for specific servers. **Associating a new server:** Click the ＋ icon to create a new server and associate it with the pool. **Removing associated servers** 1. Click the server you wan to disassociate from the pool. 2. Click the 🗑 icon. You can select multiple servers by clicking each server once. Similarly to unselect, click the respective selection once.
Shared Pool	Indicate if the pool is shared or non-shared. If shared, you can associate this pool with different application designs at different lifecycle stages. If non-shared, you can associate this pool with just one application design and lifecycle stage at any given time.
Change Image	You can change the image of the pool by clicking this

Field	Description
	link and selecting the required one from the available list of images. You can also extend the library by uploading your favorite images.

4. Click **Create**.

Modify existing pools

To modify existing pools:

1. Navigate to **Release Automation** > **Server Configuration** > **Pools**.
   The system lists all the existing pools in the right pane.

2. Click the gear next to the pool you need to modify and select **Edit**.

3. In the **Edit Pool** dialog, make the required modifications.

   > **Note:** You cannot change the 'Shared Pool' option setting in 'Edit' mode.

4. Click **Save**.

Delete existing pools

To delete existing pools:

1. Navigate to **Release Automation** > **Server Configurations** > **Pools**.
   The system lists all the existing pools in the right pane.

2. Click the gear next to the pool you need to delete and select **Delete**.

3. In the **Confirmation Required** dialog, click **Yes** to confirm deletion.

Associate server pools with lifecycle stages

You can associate a server pool with a lifecycle stage. With this, whenever an application in that lifecycle stage is deployed, the machine associated with the server pool will be used for deployment.

You can also disassociate a server pool from a lifecycle stage.

Associating a server pool with a lifecycle stage

1. Navigate to **Release Automation** > **Release Pipeline**.

2. Select the design type (**Topology Designs** or **Sequenced Designs**) from the drop-down list.

3. Select the application and the version with which you want to associate a server pool.

4. Click the gear icon next to the lifecycle stage with which you want to associate a resource pool and select **Associate Pools**.

5. In the **Select Pools for <lifecycle stage>** dialog box, select the pools you want to associate with the application version, from the list of **Available Pools** and click **Add**. The selected pools will now appear under **Selected Pools**.

6. Click **Save**.

Disassociating a server pool from a lifecycle stage

1. Navigate to **Release Automation** > **Release Pipeline**.

2. Select the design type (**Topology Designs** or **Sequenced Designs**) from the drop-down list.

3. Select the application and the version from which you want to disassociate a server pool.

4. Click the gear icon next to the lifecycle stage from which you want to disassociate a server pool and select **Associate Pools**.

5. In the **Select Pools for <lifecycle stage>** dialog box, select the pools you want to disassociate from the list of Selected Pools, and click **Remove**.

6. Click **Save**.

Use
Associate server pools with lifecycle stages

Send documentation feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

Feedback on Use (Codar 1.90)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to clouddocs@hpe.com.

We appreciate your feedback!